



Nº 9 - NOVIEMBRE 1984

315 PTAS.

NOVEDADES Y COMENTARIOS DE SONIMAG
CONCURSO ESCOLAR (PAG. 44)

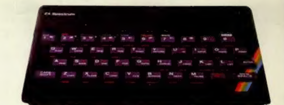
Lleva tu propia contabilidad

Arte y Ciencia de la programación
Viaje al centro de tu Commodore
Multiplicación larga en Basic
Formatación numérica

¡Hazte socio del Club de Soft de España!

Tendrás multitud de ventajas...

- Recibirás puntual información de las novedades de soft que haya en el mercado.
  Te enviaremos un catálogo bimensual que incluye más de 200 programas para Commodore 64 y Spectrum.
- Podrás adquirir programas con un 30% de descuento, por ser miembro del CLUB DE SOFT DE ESPAÑA.
- Obtendrás descuentos de hasta un 20% sobre hardware.

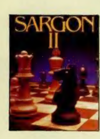


¿COMO PUEDES HACERTE SOCIO DEL CLUB DE SOFT DE ESPAÑA?

- Sólo tendrás que abonar 1.000 Ptas. como cuota UNICA de inscripción.
- Como regalo de bienvenida a nuestro CLUB de SOFT de ESPAÑA podrás elegir entre dos programas:

AJEDREZ	STARSHIP ENTERPRISE
SOLO FLIGHT	SLIPPERY SID
para CBM-64	para SPRECTRUM

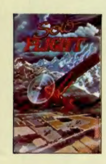
Te enviaremos el programa elegido completamente GRATIS, junto con la confirmación de tu pertenencia al CLUB de SOFT de ESPAÑA.





CLUB de SOFT de ESPAÑA





DIFUSION POR AMISTAD

Habla a tus amigos de las enormes ventajas que supone pertenecer al CLUB DE SOFT DE ESPAÑA. Si junto con tu inscripción se inscriben dos amigos tuyos por tu recomendación, te **regalamos** un programa de juegos.



CUPON DE INSCRIPCION AL CLUB DE SOFT DE ESPAÑA

Eraso, 12 - Tel. 24610 94 - 28028-MADRID

Nombre ____________________

Dirección ____________________

Ciudad ____________________ Tel. ____________________

Deseo pertenecer al CLUB DE SOFT DE ESPAÑA. Ruego me envíen como regalo de bienvenida al CLUB el programa ____________________

Firma

El CLUB DE SOFT DE ESPAÑA te enviará el programa solicitado, junto con un reembolso de 1.000 Ptas. más gastos de envío, como cuota de inscripción al Club.

DIFUSION POR AMISTAD

NOMBRE de un amigo ____________________

Dirección ____________________

Programa de regalo de bienvenida ____________________

Firma

DIFUSION POR AMISTAD

NOMBRE de un amigo ____________________

Dirección ____________________

Programa de regalo de bienvenida ____________________

Firma

Commodore
WORLD

Commodore World
está publicado por SIMSA
y la colaboración
de todos nuestros lectores

EQUIPO
Manuel AMADO; Nieves CHESA;
José Luis ERRAZQUIN;
M.ª Jesús GARCIA;
Miguel A. HERMOSELL;
Alvaro IBAÑEZ; María LOPEZ;
Juan MARQUEZ; Juan MARTINEZ;
Pere MASATS; Victoria MORALES;
Rafael PARDO; Diego ROMERO;
Albert SANGLAS; Jordi SASTRE;
Valerie SHANKS; Francisco ZABALA...
...Y NUESTROS LECTORES

SIMSA
Coordinador María López
c/Barquillo, 21-3º. Izda. - 28004
Madrid. Teléf.: (91) 231 23 88/95
Delegación en Barcelona:
Sant Gervasi de Cassoles, 39 despacho 4.
08022 Barcelona
Tels.: (93) 212 73 45/212 88 48

Colabora MEC-COMMODORE
con Club Commodore
Coordinador Pere Masats
Taquígrafo Serra, 7-5º
08029 Barcelona
Tels.: (93) 250 51 03/02



Imprime:
IBERDOS, S. A.
Germán Pérez Carrasco, 24.
28027 Madrid
Depósito Legal: M-2944-1984

SUMARIO

CONTENIDO — PAG.

PROXIMO NUMERO

PROCESO DE TEXTOS
GRAN PROGRAMA GEOGRAFICO
RUTINAS BASIC (II) - CONTROL DEL CURSOR
AYUDA PARA TECLEAR LOS PROGRAMAS
JUEGOS
CLUB COMMODORE
...Y TODAS VUESTRAS COLABORACIONES



Commodore World es miembro de CW Communications/Inc., el grupo editorial más grande y de mayor prestigio en el mundo en lo que se refiere al ámbito informático. Dicho grupo tiene a su cargo la edición de 52 publicaciones relacionadas con los ordenadores en 19 países. El grupo editorial está integrado por:
ALEMANIA: Computerwoche, Micro Computerwelt, PC Welt, Software Markt, CW Edition/Seminar, Computer Business, Run (Commodore). **ARABIA SAUDI:** Saudi Computerworld. **ARGENTINA:** Computerworld/Argentina. **AUSTRALIA:** Australasian Computerworld, Micro Computer Magazine, PC World. **BRASIL:** DataNews, Micro-Mundo. **DINAMARCA:** Computerworld/Danmark, Micro Verden. **ESPAÑA:** Computerworld/España. MicroSistemas, Commodore World. **ESTADOS UNIDOS:** Computerworld, Hot CoCo, InCider, InfoWorld, Microcomputing, PC World, 80-Micro, Mac World Micro Market Word, PC Jr. World Run (Commodore). **FINLANDIA:** Mikro. **FRANCIA:** Le Monde Informatique. **HOLANDA:** Computerworld Benelux, Micro/Info. **ITALIA:** Computerworld Italia. **JAPON:** Computerworld Japan, Perso Comworld, PC World. **MEXICO:** Computerworld/México, Compumundo. **NORUEGA:** Computerworld Norge, Mikro Data. **REINO UNIDO DE GRAN BRETAÑA:** Computer Management. Computer Business Europe. **REPUBLICA POPULAR DE CHINA:** China Computerworld. **SINGAPUR:** Asian Computerworld. **SUECIA:** ComputerSweden. MikroDatorn, Min Hemadator.

SUPER-INTERESANTISIMO

MAS SOBRE EL EXTRAORDINARIO

ACLARACIONES

Parece existir varias dudas sobre el Número Extraordinario de Programas que publicaremos cara a las próximas fiestas de Navidad y que, como dijimos ya, esperamos presentar en SIMO.

TODOS LOS LECTORES pueden reservar su copia, sean o no suscriptores —los dos llevan un descuento sobre el P.V.P. de 1.250 ptas., aunque este descuento es diferente para los suscriptores y los lectores no suscriptores.
El suscriptor con un descuento muy especial del 50% que es el precio de coste. *El lector no suscriptor* que nos envíe el boletín de reserva, tiene un 25% de descuento por lo que le queda el ejemplar en 935 ptas.

Hay algunos suscriptores que no saben su número de suscriptor. *Este número viene siempre en la etiqueta del sobre.*

De todas formas, si algún suscriptor no tiene su número a mano, puede enviar el boletín de reserva, indicando que no lo recuerda.

Transcribo a continuación la carta de un lector que ya puede ser reflejo de dudas de otros lectores:

"Aprovecho la ocasión para sugerirles, aún y cuando dicho número extraordinario ya esté confeccionado, de que el contenido del mismo, los 75 ó 100 programas que Uds. anuncian, sean al 50% de utilización para el Vic-20 y C-64 respectivamente.

Dichos programas, ¿serán una recopilación de los ya proporcionados en números anteriores?, ya que en ese supuesto no tendrían interés alguno para mí. Asimismo, supongo que los programas de referencia, estarán ya probados, ¿o tal vez no? En caso de que no funcionen, ¿deberemos esperar la salida de otro nuevo extraordinario, en el que indique los errores que pudieran haber en ese primer extraordinario?

Con el deseo de no haberles ofendido, ya que éste no es mi propósito, sino todo lo contrario, ya que mi intención no ha sido otra que la de aclarar unos puntos de los cuales nada indicaban en la nota de solicitud que incluíais en el número 8 de la revista.

Un cordial saludo.

Luis Mercade
c/ Joan Güell 82-84, 3º, 2º
Barcelona 08028
Nº de suscriptor 841

Nada hombre, no nos ofendes. Al contrario, de hecho nos es muy útil para aclarar estos puntos en beneficio de todos los demás:

1.—Este libro se pensó desde el primer momento en que fuera de similar utilidad para los usuarios del VIC, como del 64 y así está confeccionado, guardando un equilibrio entre los dos.

2.—*Todos* los programas son inéditos. Indudablemente, una recopilación de los programas publicados en Commodore World a lo largo del año tendría un interés más que limitado.

3.—Los programas que se publiquen provienen al igual que en Commodore World, de tres fuentes. Unos son nuestros, otros adaptados de nuestras revistas hermanas de todo el mundo y otros de diversos colaboradores. Todos ellos están probados y comprobados por Diego y Alvaro.

4.—Por supuesto que en el caso de que se produjera algún error o errata no es necesario esperar seis meses a otro extraordinario, estos posibles errores se rectificarían en el número de enero. Igualmente, cualquier comentario, rectificación, ampliación o mejora de los mismos, que nos enviéis, las iremos publicando en Commodore World, según vayan llegando.

5.—Para mejor comodidad de todos y, a sugerencia de otros lectores, incluimos boletín de suscripciones y de reserva encartados, para poder utilizar uno u otro (¡o los dos!) más fácilmente.

6.—Igualmente, el boletín de reserva va "más claro" para evitar confusiones.

7.—Varios lectores nos preguntan si pueden comprar más de un ejemplar. El número de reservas *NO* está limitado a un número por lector, suscriptor o no.

Si alguien desea enviar un ejemplar de regalo a un amigo o familiar, se lo podemos enviar directamente desde la redacción con tarjeta de felicitación, donde se dice quién envía el regalo y envuelto especialmente en papel de regalo.

AMIGOS COLABORADORES

Por favor, si alguno de vosotros no ha recibido noticias nuestras, no os preocupéis. Hemos estado super cargados de trabajo con todo el tema del Concurso Escolar y del Extraordinario PERO... PROMETO... que en los próximos 15 días tendréis noticias nuestras TODOS los que han enviado trabajo y, algunos de vosotros, con sorpresas interesantes ya que todos aquellos programas que se hayan incluido en el Extraordinario recibirán su remuneración correspondiente.

MERCADILLO Y PIRATEO

La sección de Mercadillo nació como una ayuda e intercambio entre amigos. Es un servicio totalmente gratuito para que los lectores puedan intercambiar experiencias, puedan vender equipos que ya no necesitan o beneficiarse de una compra más económica de 2.ª mano. En ningún momento se ha intentado ayudar y, mucho menos, potenciar o regalar publicidad gratuita al "negocio casero underground" de la copia y venta ilegal de soft.

Ultimamente ha habido serios abusos en la utilización de este Servicio, algunos descarados, enviando incluso simplemente iniciales con Apartados Postales.

Por favor, tomad nota de lo que se admitirá en Mercadillo a partir de ahora:

a) Venta de hard.

b) *Intercambio* de hard o soft.

c) Intercambio o venta de libros, revistas, literatura, etc.

d) No se admitirán anuncios de venta de soft.

e) Todas las peticiones debéis acompañarlas de Nombre completo (no iniciales); Dirección (si deseáis Apartado por comodidad de Correo especificadlo, pero poned también una dirección); Teléfono si lo queréis poner.

f) Los anuncios, saldrán solamente un mes.

g) Para evitar confusiones, malos entendidos, etc., suprimimos el servicio de anuncios por palabras a las empresas. Aquellas que deseen anunciarse pueden pedirnos las tarifas de publicidad a la redacción.

Repetimos lo que ya hemos dicho otras veces, que lamentamos tener que tomar estas medidas por culpa del abuso de media docena de listos, pero nos negamos a jugar a "censores" de los anuncios que nos enviáis.

FINALMENTE

La segunda parte de Pasaporte al Mundo, la publicaremos en nuestro próximo número debido a que en el último momento hemos visto un par de cosas que no nos convencían y que queremos comprobar mejor.

Convocatoria del Primer Concurso Nacional Escolar de Programación patrocinado por Commodore World y Microelectrónica y Control

BASES

1º Podrán participar en el mismo todos aquellos niños y jóvenes que estén en edad escolar pre-universitaria o equivalente.

2º Habrá tres grupos definidos:

a) E.G.B.
b) B.U.P.
c) F.P.

3º Los trabajos pueden ser presentados por equipos o por individuales.

4º Los trabajos presentados por equipos deben especificar:

a) Nombre del equipo.
b) Centro al que representan.
c) Nivel de enseñanza que cursan (E.G.B.-B.U.P-F.P.).
d) Nombre del profesor, monitor o coordinador.

5º Los trabajos presentados por individuales deben ir acompañados de:

a) Nombre del autor.
b) Edad y curso que estudia.
c) (¡) Si concursa representando a un centro: nombre del mismo y nombre del profesor responsable.
(¡¡) Si concursa por sí mismo: nombre del colegio donde estudia y firma del director o profesor responsable avalando que en tanto cuanto conoce al concursante el trabajo que presenta es suyo propio.

6º El tema del concurso es la realización de un programa educativo sobre materia libre a elegir por los concursantes. La extensión del programa es también de libre elección.

7º Cada centro puede presentar el número de equipos o concursantes individuales que desee, siempre que cada trabajo sea enviado por separado.

8º Los trabajos deben incluir:

a) Nombre el programa.
b) Nombre del computador para el que está confeccionado y periféricos adicionales que se necesitan (si fuese el caso).
c) Objetivo del programa.
d) Explicación detallada del programa (a máquina o impresora).
e) Listado completo a impresora.
f) Grabación en cinta o disco.

9º Todos los trabajos presentados deben ser originales e inéditos.

10º Los centros que pueden presentar equipos o concursantes individuales son:

a) Todo colegio, escuela o instituto, público o privado que imparta las enseñanzas de E.G.B., B.U.P, o F.P.
b) Centros de Informática que tengan grupos de enseñanza específica para niños y jóvenes menores de 18 años.
c) Clubs de informática que mantengan actividades específicas para niños y jóvenes menores de 18 años.

11º Los trabajos deben ser enviados antes del 31 de diciembre de 1984 a SIMSA, c/ Barquillo, 21-3º Izqda. 28004 Madrid.

12º El fallo del concurso se dará a conocer en la primera semana de marzo de 1985 en el lugar, fecha y hora que se indicarán.

13º Los trabajos premiados quedarán propiedad de los patrocinadores.

RELACION DE PREMIOS EN PAG. 44

CONCURSOS PERMANENTES

Colaboraciones de Programas y Magia

La verdad sea dicha: os estáis portando; las colaboraciones que nos están llegando son numerosas y de muy buena calidad. Por lo tanto, vamos a aclarar las condiciones de nuestros concursos permanentes.

1) Los sorteos se realizarán en los meses de junio y diciembre.

2) Se sortearán siempre seis premios, tres de 15.000, 10.000 y 5.000 pesetas en metálico de "Commodore World" y tres del mismo valor en material de Microelectrónica y Control, entre todas las colaboraciones publicadas.

3) Se sortearán siempre seis premios, tres de 3.000, 2.000 y 1.000 pesetas en metálico de "Commodore World" y tres del mismo valor en material de Microelectrónica y Control, entre todas las contribuciones publicadas en la sección de "Magia".

4) Adicionalmente, se sortearán diversos premios de cintas, juegos, etc. donados por firmas y distribuciones, entre todas las colaboraciones.

5) Los autores de las colaboraciones vendidas dentro de nuestro "Servicio de Cintas" percibirán el 20% del precio de la cinta.

6) Todas las colaboraciones deben venir escritas a máquina y los programas grabados en cinta (si es posible), o con el listado completo en impresora. Nuestros lectores más jóvenes pueden escribir a mano pero con letra muy clara.

7) Quedan automáticamente descalificadas tanto del sorteo como del "Servicio de Cintas" las colaboraciones que hayan sido enviadas a otras revistas.

8) Las colaboraciones se enviarán a "Commodore World", *C/Barquillo, 21-3 izda. 28004 Madrid.*

MAS VIAJES Y OTRO REGALO DE "IMPRESION"... A POR EL SUSCRIPTOR 8.000

Es evidente que a nuestros sucriptores les gusta viajar. De momento Joan Pujiula visitará los Estados Unidos en julio y Juan Mejuto Iglesias irá a Inglaterra en el otoño. Queríamos sortear algo un poco especial ENTRE TODOS LOS SUCRIPTORES cuando lleguemos al suscriptor número 8.000, y hemos optado por dos premios, uno concedido por Microelectrónica y Control y otro concedido por "Commodore World".

MEC regalará una impresora MPS 801 (o similar), y nosotros hemos escogido un surtido de viajes muy "chulos" para dos personas para que el ganador elija el que más le apetezca. La selección que hemos hecho es la siguiente:

Viajes de 4 días en hoteles de primera clase a uno de estos lugares: París, Lisboa, Atenas, Amsterdam, Niza, Viena.
O viajes de una semana en hoteles de 4/5 estrellas: Canarias o Mallorca.

NO OLVIDEIS NUESTRA NUEVA DIRECCION:
BARQUILLO, 21-3º IZDA.
28004 MADRID
TELEF.: 231 23 88
231 23 95

TIRADA DE COMMODORE WORLD
Certificación de 28.000 ejemplares

Según certificado firmado por la auditoría internacional DELOITTE HASKINS SELLS, S. A. el 5 de marzo de 1984, la tirada de Commodore World es de 28.000 ejemplares. Este certificado se encuentra en las oficinas de S.I.M.S.A. para comprobación de cualquier persona o entidad que lo desee.

Convocatoria del primer concurso nacional de programación en 80 columnas de Commodore-64 y 40-80 columnas del VIC-20 patrocinado por Ferre-Moret, S.A. Con la colaboración de Commodore-World, más de 500.000 ptas. en premios

BASES

1º Podrán participar todos los usuarios de COMMODORE-64 y VIC-20 que hayan obtenido la tarjeta de 80 columnas de FERRE-MORET, S.A.

2º Cada concursante podrá presentar, un programa realizado por él o por un equipo del que deberán indicar un titular.

3º Se enviará la explicación detallada del programa redactado en español, escrito a máquina. El programa se enviará en cassette o disco, adjuntando su correspondiente listado del programa.

4º Se enviarán por correo certificado a la siguiente dirección, FERRE-MORET, S.A. calle Buenos Aires nº 30, 2º-3.ª de BARCELONA-36. Antes del 31 de enero de 1985.

5º Los programas no premiados podrán ser solicitados por sus propietarios en el plazo de treinta días después del fallo.

6º El resultado se comunicará, directamente a los ganadores y se publicará en la revista COMMODORE WORLD.

7º Los premios se entregarán durante la primera semana de marzo de 1985 coincidiendo con la entrega de premios del PRIMER CONCURSO NACIONAL ESCOLAR DE PROGRAMACION.

8º Los programas premiados quedarán en propiedad de FERRE-MORET, S.A.

9º Los programas que no obtengan premio, recibirán, en caso que así lo considere FERRE-MORET, S.A., ofertas de compra de sus programas.

10º Los premios ascenderán a 500.000 Ptas. repartidos de la siguiente manera:

1º 250.000 Ptas.
2º 150.000 Ptas.
3º 100.000 Ptas.

11º Todos los participantes, que no hayan recibido premio u oferta de compra, podrán participar en el concurso de colaboradores que habitualmente COMMODORE WORLD viene realizando con un premio extra de FERRE-MORET, S.A., de 50.000 Ptas.

12º Los trabajos deberán de ser inéditos y desarrollar temas de utilidad, programas profesionales, científicos, educativos, etcétera.

FERRE-MORET, S.A. se reserva el derecho de modificar las bases de este certamen, dejarlo desierto o suprimirlo, por causa de fuerza mayor. En este caso la empresa se compromete a comunicarlo a través de COMMODORE WORLD.

Clave para interpretar los listados

Todos los listados que se publican en esta Revista han sido ejecutados en el modelo correspondiente de la gama de ordenadores COMMODORE. Para facilitar la edición de los mismos en la Revista y para mejorar su legibilidad por parte del usuario, se les ha sometido a ciertas modificaciones mediante un programa escrito especialmente para ello. Para los programas destinados a los ordenadores VIC-20 y COMMODORE 64, en los que se usan frecuentemente las posibilidades gráficas del teclado, se han sustituido los símbolos gráficos que aparecen normalmente en los listados por una serie de letras entre corchetes [] que indican la secuencia de teclas que se deben pulsar para obtener el carácter deseado. A continuación se da una tabla para aclarar la interpretación de las indicaciones entre corchetes:

[CRSRD] = Tecla cursor hacia abajo (sin SHIFT)

[CRSRU] = Tecla cursor hacia arriba con SHIFT)

[CRSRR] = Tecla cursor a la derecha (sin SHIFT)

[CRSRL] = Tecla cursor a la izquierda (con SHIFT)

[HOME] = Tecla CLR/HOME (sin SHIFT)

[CLR] = Tecla CLR/HOME (con SHIFT)

Las indicaciones [BLK] a [YEL] corresponden a la pulsación de las teclas de 1 a 8 junto a la tecla CTRL. Lo mismo sucede con [RVSON] y [RVSOF] respecto a la tecla CTRL y las teclas 9 y 10.

El resto de las indicaciones constan de la parte COMM o SHIF seguidas de una letra, número o símbolo —por ejemplo [COMM+] o [SHIFA]—. Esto indica que para obtener el gráfico necesario en el programa deben pulsarse simultáneamente las teclas COMMODORE (la que lleva el logotipo) o una de SHIFT y la tecla indicada por la letra, el número o el símbolo, en el ejemplo anterior: COMMODORE y + o SHIFT y A, respectivamente.

En los signos gráficos además se cuenta el número de veces que aparece. Por ejemplo, [7 CRSRR] equivale a 7 pulsaciones de la tecla cursor a la derecha y [3 SPC] tres pulsaciones de la barra espaciadora. ■

INDICE DE ANUNCIANTES

John KULA (RUN EE.UU.)

Traducido por Valerie SHANKS

El arte y la ciencia de la programación

La programación es tanto un arte como una ciencia. En este artículo hablamos de varias técnicas que te ayudarán a ser un programador más creativo y competente.

Antes de empezar a programar, tienes que tener en cuenta dos requisitos importantes, sin los cuales un programa puede resultar bueno pero nunca llegará a ser excepcional.

El primero de ellos es la investigación, es decir que tienes que comprender perfectamente no solamente toda la información necesaria para el futuro programa, sino también la finalidad de este.

Por ejemplo, si quieres escribir un programa de utilidades que vuelve a numerar las líneas de los programas almacenados en la memoria, también tienes que decidir (entre otras cosas) si quieres volver a numerar todos los números de línea anidados (es decir, los que siguen un GOTO, GOSUB, etc.) o si prefieres hacerlo más tarde a mano.

Las dos opciones son buenas, siempre y cuando se elige una a conciencia. Pero si no has realizado ningún tipo de investigación, y por lo tanto no te habías enterado de que existía una limitación, eso es otra cosa.

El programa que vuelve a numerar las líneas constituye un ejemplo sencillo; pero esta norma adquiere cada vez más importancia a medida que se aumenta la dificultad del programa. Actualmente estoy trabajando en un programa simulador para el C-64 que requiere un total de más de 100K de memoria (hablaré de esto más adelante). Naturalmente, el programa es complejo, y tardé tres meses en realizar mis investigaciones; pero estoy convencido de que quedan cubiertas por lo menos el 95% de las posibilidades.

El segundo paso es hacer un resumen. Algunos programadores insisten en el uso de un diagrama de flujo complejo y detallado, pero yo sospecho que esto es una manía que data de los tiempos en que los programas tenían que ser escritos en lenguaje máquina. Aunque ciertas secciones de un programa requieran un diagrama de flujo, esto es necesario solamente cuando la secuencia lógica se hace un poco complicada.

Otra técnica útil es escalonar todos los bucles For...Next, de la misma forma que se hace en el lenguaje Pascal

Es más importante para el esquema general hacer un resumen de los pasos y bloques del programa. Sin embargo, en caso de que resulte absolutamente necesario utilizar un diagrama de flujo, no hace falta de momento que te preocupes del uso correcto de los símbolos... esto se aprende sobre la marcha. Estos símbolos tienen que constituir una ayuda y no un obstáculo.

Escribir un programa

No tienes que seguir unos pasos establecidos para escribir un programa. Para tener que depurar algo, primero tienes que escribirlo. Evidentemente, si trabajas un poquito, la codificación puede resultar bastante sencilla y elegante. Pero tienes que evitar el síndrome del procesador de textos.

Los procesadores de textos te permiten realizar cambios fácilmente, y dan la sensación de que la perfección está a la vuelta de la esquina. Ocurre lo mismo con los ordenadores, sobre todo con los que disponen de un buen editor de pantalla.

Cuando tu programa funciona como tenía que funcionar sin dar ningún problema, intenta abreviarlo y luego ¡déjalo tal y como está! Cuanto más lo toques, más probabilidades tienes de que te salga un problema. Vale más saber dejarlo a tiempo que ahorrar uno o dos bytes más de memoria.

Me resulta útil salvar dos versiones de un programa terminado. La copia de tra-



bajo es la versión abreviada, para que tarde menos tiempo en cargarse y ejecutarse. La copia de reserva es la más completa y documentada, y me ayuda a seguir la lógica y acordarme de mis suposiciones si tengo que hacer unas modificaciones unos meses más tarde (y hay pocos programas que no se benefician de un par de cambios).

Documentación de un programa

Hay un par de trucos que resultan especialmente útiles para documentar un programa. Primero, utilizo la sentencia REM con frecuencia para explicar lo que está pasando en las zonas clave y para indicar cada subrutina. A menudo uso números de línea que terminan en cero para las líneas del programa, y utilizo números que terminan en cualquier otro dígito para indicar las REM o los espacios (una línea que contiene solamente dos puntos no afecta el programa, pero deja una clara separación en el listado). Este sistema de numeración me ayuda a borrar las líneas no deseadas a la hora de abreviar el programa para quedarme con la versión de trabajo.

Aquí el único problema es que la línea para una sentencia GOTO o GOSUB no debe ser una línea reservada para un comentario o un espacio. Aunque tú no tengas la intención de abreviar el programa, se lo podías dejar a una persona (como yo, por ejemplo) que sí lo va a hacer. Sé generoso con los curiosos.

Otra técnica útil es escalonar todos los bucles For...Next, de la misma forma que se hace en el lenguaje Pascal. De este modo resulta más fácil seguir la lógica del listado y ayuda a localizar cualquier error en el anidado de los bucles. Aquí de nuevo, los dos puntos entrantes sirven muy bien para escalonar; se pueden colocar uno encima del otro sin causar ningún problema.

Por último, las dos primeras líneas de cada uno de mis programas de reserva contienen información sobre los listados para la versión impresa. Todos mis listados empiezan por el nombre del fichero en letras grandes además de la fecha de la última actualización o modificación. Guardo sin tocar los listados importantes (los que incluyan modificaciones importantes y/o significativas) para poder referirme a ellos en el futuro. Los otros pueden ser usados para encender la chimenea.

Los comandos

Yo uso un cartucho de "Ayuda al Programador" que facilita en gran parte la edición de los programas en Basic al disponer de más comandos. Existen varias marcas comerciales en el mercado, cada uno con sus méritos y sus defectos.

Depuración de errores

La depuración de un programa es una experiencia desagradable que hace sufrir hasta a los profesionales. La depuración realmente tiene que ver con dos tipos distintos de problemas. Los problemas de sintaxis surgen de los errores cometidos en el uso del lenguaje Basic y normalmente son fáciles de localizar, dado que la mayoría de ellos provocan un pequeño desastre; aparece un mensaje de error en la pantalla y el programa se para; o el ordenador queda colgado y no contesta al teclado; o la pantalla lleva tanto tiempo sin cambiarse que empiezas a sospechar (la causa de este problema es un bucle sin fin, cosa que en sí no es catastrófica, pero que igualmente hace parar el programa).

El otro tipo de problema es funcional, y tiene unas consecuencias más serias dado que no lo descubres hasta que se ejecuta el programa y se reciben unas respuestas o resultados no esperados. Dado que un error de sintaxis indica el número de línea donde se pueda encontrar el error, los problemas serios deben ser más fáciles de diagnosticar y corregir.

¿Y los demás? Bien, existen básicamente dos posibilidades: o las suposiciones básicas del programa son incorrectas y la simulación de los sucesos reales no es realista, o bien existe un error en la lógica de la codificación. Vamos a suponer que estás convencido de la precisión del programa; la alternativa es bastante triste y tiene pocas posibilidades de arreglo a parte de más investigación y codificación.

La solución es la reducción de la zona de búsqueda. Primero, crea unos datos de prueba (cuanto más sencillos mejor) y escríbelos en el programa. Repasa el programa e introduce puntos de referencia (es decir, el comando Stop) en cada punto en que los datos acaban de ser manipulados.

Existe un aspecto de la programación que no recibe mucha publicidad pero que esta relacionado con la depuración y es la búsqueda de errores.

Ahora ejecuta el programa y compara los datos producidos en cada punto de referencia con los datos que tú tienes calculados a mano. Si coinciden, teclea CONT y repite todo el proceso en el siguiente punto de referencia. Cuando llegas a un punto de referencia en que los datos no coinciden, esto indica que el problema se encuentra entre este punto y el último comando Stop. Por supuesto, este proceso de depuración resulta menos laborioso si dispones de un programa de utilidades con opciones de búsqueda o de ejecución paso a paso.

Es una trampa

Existe un aspecto de la programación que no recibe mucha publicidad pero que esta relacionado con la depuración y es la búsqueda de errores, que pudieran introducirse en el sistema a través del usuario. El PET Basic, versión 2.Ø, contenido en el VIC-20 y el C-64, realiza parte de este proceso automáticamente, pero no hace lo suficiente. El Basic te pone una trampa si intentas introducir un "string" cuando estaba esperando un número, pero no al revés (obviamente no te puede leer los pensamientos para saber si querías introducir un "string" numérico o no).

Por ejemplo, uno de los grandes dolores de cabeza del PET Basic es el "string" nulo. Si la variable de "string" no contenía datos anteriores, aparece el "string" nulo cuando contestas el mensaje de entrada de "string" pulsando la tecla Return. Nosotros sabemos que allí no hay nada, pero el ordenador no se divierte mucho, especialmente si le ordenas que te busque el valor ASCII de un "string" nulo. Una solución que se usa con frecuencia es la de añadir CHR$(Ø) a cada input de "string". Por ejemplo:

```
1Ø INPUT"STRING";Q$
2Ø Q$=Q$+CHR$(Ø)
```

La teoría es que un "string" normal no quedará afectado, pero que garantizará que un "string" nulo no deseado contenga algo que pueda reconocer el ordenador. Desafortunadamente, algo que añade al "string", de modo que si en algún momento necesitas saber la longitud o el valor numérico del "string", te encontrarás con un problema más gordo que al principio.

La solución que yo sugiero es más corriente, pero por lo menos asegura que no habrá un "string" nulo ni un "string" con añadiduras. La presento a continuación:

```
1Ø INPUT"STRING";Q$
2Ø IF Q$=""GOTO 1Ø
```

El arte de detectar errores significa tener en cuenta todos los tipos de entrada posibles y juzgar su validez. Solamente porque tú especificas un rango de respuestas aceptables en tu programa no significa que alguien no probara con otra respuesta, aunque sea sin querer. Si tú esperas una respuesta numérica dentro de un rango pequeño, utiliza un comando ON...GOTO para probar la entrada. Puedes castigar los datos malos con un

mensaje duro, adjudicarles un valor poco popular o hacer cualquier otra cosa.

Lo que no hay que olvidar de los comandos ON...GOTO y ON...GOSUB es que hacen la prueba para ver si están los números enteros positivos. Los valores de cero pasan por defecto a la siguiente línea del programa, al igual que los valores mayores que las opciones que tú tienes en cuenta. Por ejemplo:

```
1Ø INPUT"NUMBER(4 TO 6)";Q
2Ø ON Q-3 GOTO 4Ø, 5Ø, 6Ø
3Ø PRINT"INTENTALO DE NUEVO,
   TONTO": GOTO 1Ø
4Ø PRINT"Q=4": END
5Ø PRINT"Q=5": END
6Ø PRINT"Q=6": END
```

La variable Q de la línea 1Ø se ha convertido en la variable Q-3 en la línea 2Ø. Si esta variable es igual a cero o mayor que el número de artículos de la lista, el ordenador ignora el comando ON...GOTO (línea 2Ø) y ejecuta la línea siguiente (línea 3Ø).

¿Qué ocurre con las variables negativas? Resultarán en un error de ?Illegal Quantity y el ordenador quedará colgado. Este ejemplo demuestra perfectamente la necesidad de tener una forma de detectar errores. La mayoría de las personas ni intentarían introducir un número negativo, y un accidente de este tipo sería muy poco probable. Pero en este ejemplo, una entrada de Ø, 1 ó 2 daría una variable negativa en la línea 2Ø. Añade el siguiente truco para pescar errores en la línea 15.

```
15 IF Q<3 THEN Q = 3
```

Por fin todas las posibilidades han sido tenidas en cuenta.

Si mantienes las rutinas de entrada lo más flexibles posible no significa necesariamente que se detectarán todos los errores, pero sí ayudará la coexistencia pacífica entre el programa y el programador. La siguiente rutina es demasiado rígida, permitiendo solamente dos respuestas.

```
1ØINPUT"LISTO PARA CONTINUAR";
  Q$
2Ø IF Q$="S"GOTO 5Ø
3Ø IF Q$="N"GOTO 1Ø
4Ø PRINT"QUE MALO ERES etc. etc.
5Ø Continúa con el programa
```

Si añades la siguiente línea, te permite más variedad (sí, bueno, vale, por qué no)

```
15 Q$=LEFT$(Q$,1)
```

O mejor todavía, dado que lo único que quieres hacer es retrasar la ejecución del programa hasta que tú estés listo, tienes suficiente con lo siguiente:

```
1ØINPUT"LISTO PARA CONTINUAR";
  Q$
2Ø IF LEFT$(Q$,1)="N"GOTO 1Ø
```

Para que te resulte aún más fácil (la verdad es que poca gente se molesta en teclear N, esperaría hasta que estuviera lista y teclearía S), incluye la propuesta más probable dentro del mismo mensaje, para que sólo tengas que pulsar la tecla return. A continuación se presenta la codificación para una nueva línea 1Ø:

```
INPUT "SIGO [2 SPC][RVSON] S
[RVSOFF][3 CRSRL]";Q$
```

Si tecleas y ejecutas esta línea, una "S" intermitente tiene que aparecer debajo del cursor, detrás del Input. Sólo tienes que pulsar la tecla Return para activar la respuesta S. En este caso, una solución aún más elegante sería la siguiente, que es la que yo utilizo normalmente en estas circunstancias.

```
1ØPRINT"PULSA CUALQUIER TECLA
  PARA CONTINUAR"
2Ø WAIT 197,64: WAIT 197,64,64
3Ø POKE 198,Ø
```

Por si lo quieres saber, la línea 3Ø borra el buffer del teclado, para que al pulsar la tecla para continuar el programa, la propia tecla no aparezca en la pantalla provocando un error.

Sacando el jugo

Abreviar un programa es aconsejable en cualquier caso, pero si te encuentras con un error de ?Out Of Memory, este proceso se hace imprescindible. Un buen sitio para empezar es con los comentarios, sobre todo teniendo en cuenta que la línea entera de una sentencia REM se almacena en la memoria exactamente de la misma forma en que tú la tecleas, aquí no estamos hablando de la conversión en números ni el uso de los "tokens".

Otro sistema para ahorrar espacio es eliminar todos los espacios en blanco no necesarios, pero hay que tener un poco de cuidado con esto. A veces es necesario dejar un poco de espacio para que el ordenador no piense que reconoce una palabra clave. Por ejemplo, ON T AND S GOTO 13Ø puede ser abreviado a ONT ANDSGOTO13Ø. Es necesario dejar el espacio en blanco para que TAN no se considere como una palabra clave.

Hay muchos trucos como este para recuperar unos cuantos bytes. Pero cuando esto no es suficiente, la solución sería dividir un programa largo en dos o más programas cortos y cargarlos por separado. Sin embargo, esto sólo funciona cuando cada parte del programa se puede ejecutar independientemente de los demás, de modo que hay que tener cuidado.

Aunque introduzcas un comando Load para que se ejecute cuando el programa llega a este punto, habrá dificultades. No

Lo que no hay que olvidar de los comandos ON...GOTO y ON...GOSUB es que hacen la prueba para ver si están los números enteros positivos.



se puede borrar el contenido anterior sin también borrar la instrucción Load. Y si el nuevo programa es más largo que el original, te encontrarás con un problema en el momento en que el programa nuevo intente cargarse en la zona reservada antes para las variables. La solución fácil es engañarle al ordenador para que piense que está ejecutando un comando Load procedente del teclado; el comando Load recibido en el modo inmediato también ejecuta automáticamente el comando New.

Para poder hacer esto, primero tienes que fijar los comandos apropiados en la pantalla, en sus posiciones correctas, a través de las sentencias Print. De esta forma los 13 del código ASCII (Returns) se introducen mediante un Poke en el buffer del teclado. Estos Returns están cronometrados para que coincidan con la situación del cursor en la misma línea de la pantalla que una de las líneas que aparecía antes en el mismo sitio. Una vez ejecutado el primer comando, Load, el cursor se desplaza automáticamente a la segunda línea, RUN. El segundo Return del buffer ejecuta esta línea, y el segundo programa habrá sido cargado y ejecutado correctamente. Suponiendo que el nuevo nombre del programa es "NUEVO PROG", el nuevo listado tendrá la siguiente forma:

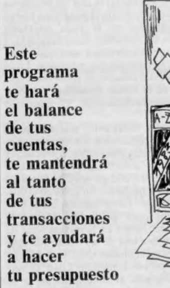
```
1Ø PRINT"[4 CRSR DNs]"
2Ø REM [CRSR DN]=CURSOR ABAJO
3Ø PRINT"LOAD"CHR$(34)"Ø:NUEVO
   PROG"CHR$(34",8[4 CRSR DNs]"
4Ø PRINT"RUN"
5Ø PRINT"[9 CRSR UPs]"
6Ø REM [CRSR UP]=CURSOR ARRIBA
7Ø POKE 631, 13:POKE 632, 13
8Ø REM ESTOS SON DOS (RETURNS)
   INTRODUCIDOS EN EL BUFFER
   DEL TECLADO MEDIANTE UN
   POKE
9Ø POKE 198,2
99 REM ESTO ESPECIFICA QUE EL
   BUFFER CONTIENE ACTUALMEN-
   TE DOS CARACTERES
```

Por último, un hecho curioso y poco conocido acerca de los ordenadores. Los chips de un circuito integrado funcionan a base de humo. Si llevas mucho tiempo delante del ordenador y ves que le empieza a salir humo, es que los chips se están quedando sin humo y por lo tanto dejarán de funcionar.

John SCHMOYER (RUN EEUU)

Traducido por Valerie SHANKS y adaptado por Alvaro IBAÑEZ

Lleva tu propia contabilidad





Este programa te hará el balance de tus cuentas, te mantendrá al tanto de tus transacciones y te ayudará a hacer tu presupuesto

Algunos de los anuncios que se ven hoy en día acerca de los ordenadores intentan convencerte de que tienes que comprarte una de estas máquinas maravillosas porque realmente te es imprescindible.

Te dicen que te tienes que mantener en tanto en cuanto a los avances tecnológicos. Te lo debes a ti y a la seguridad de tu familia comprarte el ordenador XYZ. Si no por otra cosa, ¡piensa en tus niños! Tienen que estar familiarizados con los ordenadores simplemente para poder sobrevivir la edad de la informática.

Pero ¿qué ocurre con esas miles de personas que "piensan en sus niños", y se encuentran con que los niños no hacen otra cosa que jugar a los marcianitos con el "bicho"? Por ejemplo, mi hijo de nueve años tiene muy poco interés en hacer programas en su C-64, pero le encanta tirarse todo el día jugando. Menuda gracia para mí y para miles de personas como yo que han invertido su dinero en un ordenador personal.

Casi todo el software disponible en el mercado son juegos. A mí también me gusta jugar, pero tengo mi lado práctico, especialmente cuando se trata de algo que me ha costado mucho dinero. Afortunadamente, cada día hay más software con fines más prácticos.

En este artículo, me gustaría compartir con vosotros lo que yo considero un programa práctico.

C-64

Lo que hace el programa

El programa "Contabilidad" te ayudará a seguir la pista de todas tus actividades financieras, y va más allá de simplemente registrar toda la información. Te permite sacar en la pantalla (o en la impresora si tienes una) una lista completa de tus transacciones. Si tienes todas tus actividades registradas en un par de páginas reduce el tiempo que pierdes en sacar sentido del saldo de tu cuenta bancaria. Pero aún hay más.

¿Y tu presupuesto doméstico? Quizás tengas problemas en recordar a dónde te ha ido todo el dinero. Quizás es la hora de saber exactamente cuánto se gasta en comida, ropa, transporte y otras cosas. Este programa también te ayudará en este aspecto.

El funcionamiento del programa

Para empezar, el programa te pide un nombre para el fichero. Esta característica te permite registrar toda la actividad de un mes en un fichero. Puedes guardar el fichero todo el tiempo que quieras. Lo único que tienes que hacer es asignarle un nombre nuevo a cada mes. (Yo siempre le pongo el nombre del mes.) Por ejemplo, si estamos en el mes de abril, y quieres saber lo que hiciste en marzo, introduce marzo como el nombre del fichero y lo podrás saber.

Si introduces un nombre del fichero que no se encuentra en el disco, el programa te contesta, "¿Es un fichero nuevo?". Tú tienes que contestar S o N. Esto proporciona un poco de seguridad por si te equivocas a la hora de teclear un nombre de fichero. Si el fichero se encuentra en el disco, la información se carga en la memoria.

CONTABILIDAD 64

LIDER EN VENTAS

VERSION A
300 cuentas
3 000 apuntes

VERSION B
600 cuentas
2 300 apuntes

- Listado de diario
- Balance de situación y de Sumas y Saldos
- Estractos de cuenta
- Cuenta de explotacion diario de cierre
- Mantenimiento de ficheros utilitario

EQUIPO NECESARIO
- COMMODORE 64 y unidad de disco
- IMPRESORA MPS 801 (versión A y B)
- IMPRESORAS CENTRONICS (versión B)

ADAPTADO AL PLAN GENERAL CONTABLE ESPAÑOL

P V P VERSION A 24 550
VERSION B 29 500



EXTENSA GAMA DE PROGRAMAS

PROGRAMAS EN CARTUCHO



VIZAWRITE 64

PRACTICALC 64

VIDEOCLUBS 64

1.800 videofilms
650 clientes por disco
Saldo de cada cliente
Altas y bajas videofilms
Busqueda videofilms/clientes.

P.V.P. 35.000,-

ADMINISTRACION DE FINCAS

30 inmuebles
400 inquilinos
Listado recibos
Propiedad vertical
Comunidades de propietarios

P V P 35 000,

PROGRAMAS EN CASSETTE

CONTABILIDAD PERSONAL
CONTABILIDAD DOMESTICA
PERT
CALCULOS FINANCIEROS
CALCULOS ESTADISTICOS
FICHERO MONEDAS
FICHERO SELLOS
FICHERO BIBLIOTECA
FICHERO AGENDA
QUINIELAS



casa de software s.a.

c aragón, 272, 8 °, 6 ª tel. 215 69 52 08007 barcelona

DESEO RECIBIR INFORMACION:

Nombre y dirección:

A continuación, aparece el menú principal. Tiene el siguiente aspecto:

MENU CONTABILIDAD
1=Añadir Transacciones
2=Modificar Transacciones
3=Imprimir Registros
4=Consultar sobre el Balance
5=Análisis del Presupuesto
6=Cancelar

Tienes seis opciones. Las dos primeras te permiten añadir nuevas transacciones o modificar las existentes. La tercera te permite imprimir un registro de todas las actividades en el fichero. Esto se puede hacer en la pantalla o en la impresora.

La opción cuatro calcula y saca en pantalla el saldo de tu cuenta. Esta es la forma más sencilla y rápida de saber el dinero que tienes disponible en tu cuenta.

La opción cinco te permite analizar los gastos de tu familia dentro de cada categoría presupuestaria que tú has establecido. También calculará el porcentaje que esta cantidad representa en comparación con el gasto total de un mes. La sexta opción da fin al programa.

Cómo funciona el programa

Vamos a estudiar el programa línea por línea para ver cómo funciona.

Línea 100.—Fija un límite de 150 como el máximo número de transacciones en el programa. Yo normalmente tengo 75 transacciones en un mes, por lo que creo que 150 es suficiente. Este número puede ser aumentado, pero ten ciudado, te podrías quedar sin memoria.

Líneas 110-175.—Definen varios códigos de control y colores de la pantalla y los caracteres. Puedes cambiar los colores cuando quieras.

Líneas 180-185.—Definen una variable de 40 espacios, y otra de 80 signos de igual a (=); éstos se utilizan a lo largo del programa.

Línea 190.—Inicializa los colores de la pantalla y los bordes.

Líneas 200-210.—Definen las variables de los "arrays" que contendrán las transacciones.

DT$ es la fecha de la transacción. Utiliza el formato estándard de MM/DD/AA cuando introduces la fecha; BC$ es la categoría del presupuesto, que puede ser cualquier cosa, siempre y cuando no tenga una longitud superior a los tres caracteres; DS$ es la descripción de la actividad, de hasta veinte caracteres de longitud; CK$ es el número de talón, de hasta cuatro caracteres de longitud; CA$ es el importe de talon (o el importe de los cargos bancarios, recargos, etc.); DA$ es la cantidad depositada.

Las transferencias realizadas de tu cuenta corriente a tu cuenta de ahorros tienen que ser entradas como una cantidad negativa en la cuenta corriente, para que el cuadro general de gastos no se quede descompensado. Se calcula el gasto total sumando todas las cantidades de talón.



Líneas 220-300.—Inicializan el color de carácter y sacan en pantalla un mensaje que identifica el programa.

Líneas 400-880.—Te piden el nombre del fichero. Si no contestas, o si tecleas FIN, el programa se da por terminado. El programa intenta abrir el fichero en el disco. Si no encuentra el fichero, aparece el mensaje de la línea 520, y el ordenador te pide que le confirmes que quieres crear un nuevo fichero con dicho nombre. Sin embargo, el fichero no se crea hasta más tarde, después de haberse introducido todas las transacciones.

Si se encuentra el fichero, las transacciones se introducen mediante lectura en las variables de "arrays" definidas en las líneas 200 y 210. La variable "LINES" lleva la cuenta del número de transacciones almacenadas en la memoria.

Líneas 1000-1220.—Presentan el menú principal y aceptan tu opción. A continuación el programa realiza la función elegida. La opción da por finalizado el programa.

Líneas 2000-2440.—Esta es la rutina que suma las transacciones.

Línea 2000.—Se encarga de que no se introduzcan más transacciones que el máximo número permitido (definido en la línea 100).

Línea 2010.—Borra la pantalla y saca en pantalla el número de la nueva transacción.

Líneas 2020-2130.—Representa los mensajes.

Línea 2130.—Verifica las variables DR, que está fijada en 1 solamente en la rutina de modificación. Significa "mensajes en pantalla", y vuelta a la rutina de modificaciones".

Línea 2180.—Realiza una función enormemente útil. Si se encuentra por lo menos una transacción en el fichero, la fecha de la nueva transacción pasa por defecto a la fecha de la última transacción introducida. Esto ahorra tiempo y pulsaciones, dado que se puede pagar con varios talones en un solo día.

Líneas 2190-2200.—Aceptan los datos. Si se introduce FIN, las transacciones se salvan en disco, y se vuelve al menú principal.

Líneas 2220-2360.—Aceptan la información que queda. Esta información se introduce mediante una rutina especial de entrada, empezando en la línea 5000. Esta rutina demuestra un mensaje subrayado, que indica la longitud máxima de los datos. Si no se introduce nada para una información determinada, se muestra un sólo carácter subrayado para indicar que no se ha producido ninguna entrada. Esta información quedará en blanco, pero esto no tiene por qué despistarte.

Además, cuando entras la descripción de una transcripción, no utilices la coma. El lenguaje Basic piensa que las comas sirven de separación de los datos, y te contestará con un mensaje de EXTRA IGNORED si introduces una coma.

Línea 2400.—Incrementa por uno la cuenta de las transacciones.

Líneas 2410-2430.—Te preguntan si las entradas están bien o no. Si contestas N, el programa avanza hasta la rutina de edición, que empieza en la línea 2540.

Si tienes que modificar una entrada en este momento, hazlo con cuidado. Por ejemplo, si necesitas borrar unos datos tienes que usar la barra de espacio con la tecla Shift, los espacios normales no funcionan. Parece que esto es una pega del intérprete Basic. Si te queda alguna duda, utiliza la barra de espacio con Shift, y no tendrás ningún problema.

Línea 2440.—Borra la bandera de edición y vuelve a la línea 2000, para que puedas entrar otra transacción.

Líneas 2500-2740.—Esta rutina es la que edita las transacciones.

Línea 2500.—Te pide el número de la transacción. Si introduces un cero, las transacciones se salvan en disco y el programa vuelve al menú principal.

Línea 2520.—Se encarga de que el número de transacciones introducido sea válido. La línea 2530 fija la variable DP y realiza la rutina de representación en pantalla en la línea 2020 de la rutina de suma.

Líneas 2540-2670.—Sacan la información en pantalla y te dan la opción de modificarla.

Líneas 2700-2720.—Te preguntan si los datos son correctos. Si contestas N, tendrás la oportunidad de repasar los datos, y modificar lo necesario. Recuerda que tienes que utilizar la barra del espacio con Shift para borrar los datos.

Línea 2730.—Verifica la variable EDIT, que no se fija hasta llegar a la rutina de sumar. Si se fija, el programa vuelve a la rutina de sumar. De no ser así, el programa vuelve a la línea 2500 para que puedas modificar más transacciones.

Líneas 3000-3670.—Te permiten imprimir el registro de comprobación.

Líneas 3000-3100.—Representan un menú sencillo, que te ofrece la opción de ver los datos en pantalla o de imprimirlos en la impresora, o volver al menú principal.

algunos de nuestros programas para los ordenadores personales

commodore



25.000pts.



15.000pts.



25.000pts.



8.000pts.



15.000pts.



5.000pts.



10.000pts.



10.000pts.



15.000pts.





5.000pts.



5.000pts.

B.M.

BASIC MICRO-ORDENADORES, S.A.

AVD. CESAR AUGUSTO, 72
Tlfs. 23 56 82 y 22 65 44
50003-ZARAGOZA

Línea 312Ø.—Inicializa el contador de líneas en cero y fija la variable Flag, que significa que ésta es la primera página.

Línea 3140.—Verifica la opción que ha sido elegida; si seleccionas la opción de pantalla, el programa avanza a la línea 32ØØ.

Líneas 316Ø-318Ø.—Presentan un mensaje en pantalla y abren el canal de la impresora en caracteres mayúsculas y minúsculas.

Líneas 32ØØ-329Ø.—Son utilizadas tanto por la opción de pantalla como por la de la impresora.

Línea 32ØØ.—Fija el balance original en cero.

Línea 321Ø.—Inicia un bucle por todas las transacciones.

Líneas 322Ø-323Ø.—Editan las cantidades de los talones y en la cuenta para imprimirlas.



Línea 324Ø.—Comprueba el final de página o pantalla y salta a la línea 35ØØ, si es necesario.

Línea 3245.—Verifica la variable Answer$, que podría ser fijada en "Q" al final de la rutina de pantalla en la línea 354Ø, donde tienes la opción de terminar la representación en pantalla.

Línea 3250.—Actualiza el nuevo balance restando el importe de los talones y sumando el importe de la transacción en la cuenta.

Línea 3260.—Realiza la rutina que empieza en la línea 38ØØ, con la finalidad de la impresión, edita el nuevo balance en una variable llamada NB$.

Líneas 327Ø-328Ø.—Realizan las rutinas empezando en la 33ØØ ó 34ØØ, respectivamente, para imprimir la transacción.

Línea 329Ø.—Fin del bucle.

Una vez procesadas todas las transacciones, se cierra la impresora, si éste es el caso, o si no, te indica que pulses la tecla Return. El programa vuelve al menú principal.

Líneas 33ØØ-333Ø.—Representan una sola transacción en la pantalla, y se decrementa por dos el contador de líneas. Cada transacción ocupa dos líneas.

Líneas 34ØØ-347Ø.—Imprimen una sola transacción en la impresora. Primero,

```
1 GOTO100
2 !================================
==!
3 !
  !
4 ! PROGRAMA:CONTABILIDAD: 2-1-84
  !
5 ! AUTOR : JOHN SCHMOYER,RUN EEUU
  !
6 !
  !
7 !================================
==!
8 !
9 !
100 MAXSIZE=150 :REM MAX. DE TRANSAC
CIONES
110 CSCREEN=53280:REM PANTALLA DE CO
LOR
120 CHARS$  = "[WHT]"   :REM COLOR=B
LANCO
130 SCREEN=0:REM COLOR DE LA PANTALL
A=NEGRO
140 HOME$="[HOM]":REM [CURSOR HOME]
150 CLSCREEN$="[CLR]" REM BORRADO DE
 PANTALLA
160 LOWERCASE$=CHR$(14)
170 UPPERCASE$=CHR$(142)
175 CR$=CHR$(13)
180 SPACE$="[40SPC]":REM 40 ESPACIOS

185 FORN=1TO8:EQUAL$=EQUAL$+"=======
===":NEXTN
190 POKECSCREEN,SCREEN:POKECSCREEN+1
,SCREEN
200 DIMDT$(MAXSIZE),BC$(MAXSIZE),DS$
(MAXSIZE),CK$(MAXSIZE)
210 DIMCA$(MAXSIZE),DA$(MAXSIZE)
220 PRINTCLSCREEN$;LOWERCASE$:PRINTC
HARS$
230 PRINTHOME$;"[RVSON]";SPACES$;
240 PRINT"[RVSON][SPC][RVSOFF][14SPC]
[SHIFTC]ONTABILIDAD[12SPC][RVSON][SPC]
[RVSOFF]";
250 PRINT"[RVSON]";SPACES$;
260 FORI=4TO23
270 PRINT"[RVSON][SPC][RVSOFF]";LEFT
$(SPACES$,38);"[RVSON][SPC][RVSOFF]"
;
280 NEXTI
```

```
290 PRINT"[RVSON][6SPC][SHIFTP]ULSA[SPC]
UNA[SPC]TECLA[SPC]PARA[SPC]EMPEZAR[6SPC]
";
300 GETA$:IFA$=""THEN300
400 PRINTCLSCREEN$;
410 PRINT"[RVSON][SPC][SHIFTI]NTRODU
CE[SPC]EL[SPC]NOMBRE[SPC]DEL[SPC]FIC
HERO[SPC]"
420 PRINT"[RVSOFF]-------------------
----------------"
430 CFILE$="":INPUTCFILE$
440 IFCFILE$=""ORCFILE$="END"THEN990
0
460 OPEN15,8,15,"I0":OPEN5,8,5,CFILE
$+",S,R"
480 INPUT#15,EN,EM$,ET,ES
500 IFEN<20THEN700
520 PRINT:PRINT"[2SPC][SHIFTH]MMMM,[SPC]
ES[SPC]UN[SPC]FICHERO[SPC]NUEVO";:IN
PUTNW$
540 IFNW$="S"ORNW$="SI"THEN600
560 CLOSE15:CLOSE5
580 GOTO400
600 LINES=0
620 GOTO1000
700 PRINT:PRINT"[2SPC][SHIFTL]EYENDO
[SPC]EL[SPC]FICHERO...[SHIFTE]SPERE"

710 LINES=0
720 INPUT#5,QA$,QB$,QC$,QD$,QE$,QF$
740 RS=ST
760 LINES=LINES+1
780 DT$(LINES)=QA$:BC$(LINES)=QB$
800 DS$(LINES)=QC$:CK$(LINES)=QD$
820 CA$(LINES)=QE$:DA$(LINES)=QF$
860 IFRS=0THEN720
880 CLOSE5
1000 PRINTCLSSCREEN$
1020 PRINT"[RVSON][2CRSRD][13SPC][SHIFTM]
ENU[2SPC][SHIFTC]ONTABILIDAD[9SPC][RVSOFF]
"
1040 PRINT:PRINT:PRINTTAB(8);"1[SPC]
=[SPC][SHIFTA]NADIR[SPC]TRANSACCIONE
S"
1060 PRINT:PRINTTAB(8);"2[SPC]=[SPC]
[SHIFTM]ODIFICAR[SPC]TRANSACCIONES"
1100 PRINT:PRINTTAB(8);"3[SPC]=[SPC]
[SHIFTI]MPRIMIR[SPC]REGISTROS"
1120 PRINT:PRINTTAB(8);"4[SPC]=[SPC]
[SHIFTC]ONSULTA[SPC]SOBRE[SPC]EL[SPC]
BALANCE"
```

cada dato está rodeado con el número correcto de espacios y se almacena en las variables provisionales de A$ a F$, para que cada dato aparezca en su columna correspondiente. La línea 3460 decrementa el contador de líneas por uno después de imprimir los datos.

Líneas 3500-3670.—Aquí termina la rutina Página/Pantalla. La línea 3500 comprueba para ver si la impresora fue elegida. Si así fue, el control se transfiere a la línea 3600. De no ser así, la línea 3505 verifica la variable Flag. Si ésta se ha fijado en 1, significa que ésta es la primera vez que se ejecuta esta rutina, y no hace falta pulsar la tecla Return, y el control vuelve a la línea 3240. La línea 3510 te indica que pulses la tecla Return, o que pulses la tecla Q para salir.

Si pulsas la tecla Q, el control vuelve a la línea 3240. De no ser así, la rutina borra la pantalla, representa el título, fija el contador de líneas en 24 y vuelve a la línea 3240. Las líneas 3600 a 3670 contienen la rutina de fin de página para la impresora.



Línea 3600.—Verifica la variable Flag, igual que la línea 3505, y salta el FORM FEED si se ha fijado. De esta forma no te gastas una hoja de papel la primera vez que se ejecuta esta rutina.

Línea 3605.—Envía el comando de principio de página a la impresora. Si la impresora no soporta este comando, o si no dispone de formatos continuos, necesitarás alguna codificación adecuada.

Resulta bastante sencillo simular el principio de página si imprimes el número correcto de líneas en blanco que obliguen a la impresora a avanzar hasta el principio de la página siguiente.

Líneas 3610-3640.—Imprimen la cabecera del informe.

Línea 3650.—Fija el contador de líneas en 55.

Líneas 3660.—Fija la variable Answer$ en nulo. Answer$ se puede fijar en "Q" solamente en la rutina de representación en pantalla, y no es aplicable a la rutina de impresora. Aquí se fija en nulo, dado que la línea 3245 lo comprobará cuando sales de esta rutina.

Líneas 3800-3880.—Esta rutina toma el valor de la variable NBAL y crea una nueva variable llamada NB$. A continua-

```
1125 PRINT:PRINTTAB(8);"5[SPC]=[SPC]
[SHIFTA]NALISIS[SPC]DEL[SPC]PRESUPUE
STO"
1130 PRINT:PRINTTAB(8);"6[SPC]=[SPC]
[SHIFTC]ANCELAR"
1140 PRINT"[4CRSRD][8SPC][SHIFTI]NTR
ODUZCA[SPC]SU[SPC]OPCION";:INPUT OP
1160 IF OP = 6 THEN CLOSE 15 CLOSE 5
:GOTO9900
1180 IFOP<1OROP>5THEN1000
1200 ONOPGOSUB2000,2500,3000,4300,45
00
1220 GOTO1000
2000 IFLINES+1>MAXSIZETHENRETURN
2010 PRINTCLSCREEN$;"[RVSON][SHIFTT]
RANSACCION[SPC][SHIFTN]UMERO [SPC]";
LINES+1,"[RVSOFF]"
2020 PRINT"[CRSRD][RVSON][SHIFTF]ECH
A.........[RVSOFF]"
2030 PRINT"[RVSON][14SPC][RVSOFF]"
2040 PRINT"[RVSON]CAT.DE[SPC]PRESUP.
[RVSOFF]"
2050 PRINT"[RVSON][14SPC][RVSOFF]"
2060 PRINT"[RVSON][SHIFTD]ESCRIPCION
[SPC]..[RVSOFF]"
2070 PRINT"[RVSON][14SPC][RVSOFF]"
2080 PRINT"[RVSON][SHIFTC]HEQUE[SPC]
N[COMMV].....[RVSOFF]"
2090 PRINT"[RVSON][14SPC][RVSOFF]"
2100 PRINT"[RVSON][SHIFTC]ANTIDAD[SPC]
[SHIFTC]HEQ.[RVSOFF]"
2110 PRINT"[RVSON][14SPC][RVSOFF]"
2120 PRINT"[RVSON][SHIFTS]ALDO[SPC][SHIFTA]
CTUAL..[RVSOFF]"
2130 IF DP=1THENRETURN
2180 IFLINES<>0THENPRINTHOME$,"[2CRSRD]
[14CRSRR]?[SPC]";DT$(LINES)
2190 PRINTHOME$;"[2CRSRD][14CRSRR]";
:INPUTE$
2200 IFE$=""ORLEFT$(E$,3)="FIN"THENG
OSUB4000:RETURN
2210 DT$(LINES+1)=E$
2220 LN=5:CO=15 SZ=3 GOSUB5000
2230 IFE$="[COMM@]"THENE$=""
2240 BC$(LINES+1)=E$
2250 LN=7:SZ=20:GOSUB5000
2260 IFE$="[COMM@]"THENE$=""
2270 DS$(LINES+1)=E$
2280 LN=9:SZ=4:GOSUB5000
2290 IFE$="[COMM@]"THENE$=""
2300 CK$(LINES+1)=E$
```

```
2310 LN=11 SZ=9 GOSUB5000
2320 IFE$="[COMM@]"THENE$=""
2330 CA$(LINES+1)=E$
2340 LN=13:SZ=9:GOSUB5000
2350 IFE$="[COMM@]"THENE$=""
2360 DA$(LINES+1)=E$
2400 LINES=LINES+1
2410 PRINT"[HOM][16CRSRD][SHIFTE]S[SPC]
CORRECTO?";
2420 GETA$:IFA$<>"S"ANDA$<>"N"THEN24
20
2430 IFA$="N"THENQN=LINES EDIT=1 GOS
UB2540
2440 EDIT=0:GOTO2000
2500 PRINTCLSCREEN$;"[RVSON][SHIFTN]
UMERO[SPC]DE[SPC]TRANSACCION?[SPC]0[3CRSRL]
";:INPUTQN
2510 IFQN=0THENGOSUB4000:RETURN
2520 IFQN<0ORQN>LINESTHEN2500
2530 DP=1 GOSUB2020 DP=0
2540 PRINTHOME$;"[2CRSRD]";
2550 PRINT"[14CRSRR]?[SPC]";DT$(QN)
2560 PRINT"[CRSRD][14CRSRR]?[SPC]",B
C$(QN)
2570 PRINT"[CRSRD][14CRSRR]?[SPC]",D
S$(QN)
2580 PRINT"[CRSRD][14CRSRR]?[SPC]";C
K$(QN)
2590 PRINT"[CRSRD][14CRSRR]?[SPC]";C
A$(QN)
2600 PRINT"[CRSRD][14CRSRR]?[SPC]";D
A$(QN)
2610 PRINTHOME$,"[2CRSRD]",
2620 PRINT"[14CRSRR]", INPUTDT$(QN)
2630 PRINT"[CRSRD][14CRSRR]", INPUTB
C$(QN)
2640 PRINT"[CRSRD][14CRSRR]";:INPUTD
S$(QN)
2650 PRINT"[CRSRD][14CRSRR]";:INPUTC
K$(QN)
2660 PRINT"[CRSRD][14CRSRR]";:INPUTC
A$(QN)
2670 PRINT"[CRSRD][14CRSRR]";:INPUTD
A$(QN)
2700 PRINTHOME$;"[16CRSRD][SHIFTE]S[SPC]
CORRECTO?";
2710 GETA$:IFA$<>"S"ANDA$<>"N"THEN27
10
2720 IFA$="N"THEN2540
2730 IFEDIT=1THENRETURN
2740 GOTO2500
```

ción las líneas 381Ø a 383Ø examinan NB$, buscando un punto decimal. Si éste se encuentra, el control se transmite a la línea 385Ø. Si no se encuentra un punto decimal, "".ØØ" se agrega a NB$, y se salta a la línea 386Ø. La línea 385Ø añade un Ø adicional a NB$, si es necesario. La línea 386Ø rodea NB$ de espacios para que adquiera una longitud de nueve caracteres, la longitud correcta para imprimirse.

Líneas 4ØØØ-412Ø.—Esta rutina salva las transacciones en disco. La línea 4Ø3Ø crea un nuevo fichero vacío para escribir. Las líneas 4Ø4Ø a 4Ø6Ø repasan las transacciones y las escriben en el fichero.

Líneas 43ØØ-438Ø.—Repasan las transacciones, restándoles todos los importes



de los talones y sumándoles las cantidades depositadas para poder calcular el balance final. El balance se representa en la línea 435Ø.

Líneas 45ØØ-473Ø.—La rutina del análisis del presupuesto. Esta rutina te pide la categoría del presupuesto que quieras analizar en la línea 453Ø. Al fijar E$ en "fin" antes de entrar los datos, la respuesta por defecto es "fin". De modo que si pulsas la tecla Return sin haber entrado ningún dato, el programa sabe que quieres que termine, y te devuelve al menú principal.

Línea 455Ø.—Fija NBAL (el total de todos los talones) en cero. También fija en cero ACCUM, la cantidad del dinero gastado en esta categoría.

Líneas 457Ø-46ØØ.—Repasan las transacciones, acumulando NBAL y ACCUM.

Línea 461Ø.—Le quita el signo negativo

```
3000 PRINTCLSCREEN$,"[5CRSRD]"
3020 PRINTTAB(10),"P[SPC]=[SPC]MUEST
RA[SPC]EN[SPC]PANTALLA"
3030 PRINTTAB(10);"I[SPC]=[SPC]SALID
A[SPC]A[SPC]IMPRESORA"
3040 PRINTTAB(10),"Q[SPC]=[SPC][SHIFTV]
OLVER[SPC]AL[SPC]MENU"
3050 PRINT"[2CRSRD]";TAB(10);"[SHIFTI]
NTRODUZCA[SPC]SU[SPC]OPCION",:INPUTS
Q$
3080 IFSQ$<>"P"ANDSQ$<>"I"ANDSQ$<>"Q
"THEN3000
3100 IFSQ$="Q"THENRETURN
3120 LCNT=0:FLAG=1
3130 ANSWER$=""
3140 IFSQ$="P"THEN3200
3160 PRINTCLSCREEN$;"[5CRSRD][10SPC]
[SHIFTI]MPRIMIENDO..."
3180 OPEN4,4,7:CMD4
3200 NBAL=0
3210 FORN=1TOLINES
3220 A$=RIGHT$("[10SPC]"+CA$(N),9)
3230 B$=RIGHT$("[10SPC]"+DA$(N),9)
3240 IFLCNT<5THENGOSUB3500
3245 IFANSWER$="Q"THENRETURN
3250 NBAL=NBAL-VAL(CH$(N))+VAL(DA$(N
))
3260 GOSUB3800
3270 IFSQ$="P"THENGOSUB3300
3280 IFSQ$="I"THENGOSUB3400
3290 NEXTN:IFSQ$="I"THENPRINT#4,"[SPC]
":CLOSE4:RETURN
3292 PRINT"[SHIFTP]ULSA[SPC][SHIFTR]
[SHIFTE][SHIFTT][SHIFTU][SHIFTR][SHIFTN]
",:INPUTANSWER$
3295 RETURN
3300 PRINTDT$(N),TAB(9),BC$(N),TAB(1
3);CK$(N);TAB(18);A$;"[3SPC]",B$
3310 PRINTTAB(5),DS$(N),TAB(30),"[RVSON]
",NB$,"[RVSOFF]"
3320 LCNT=LCNT-2
3330 RETURN
3400 C$=LEFT$(DT$(N)+"[10SPC]",10)
3410 D$=LEFT$(BC$(N)+"[5SPC]",5)
3420 E$=LEFT$(CK$(N)+"[6SPC]",6)
3430 F$=LEFT$(DS$(N)+SPACE$,24)
3440 PRINTC$;D$;E$;F$;A$;"[2SPC]";B$
;"[2SPC]";NB$
3460 LCNT=LCNT-1
3470 RETURN
3500 IFSQ$="I"THENGOTO3600
3505 IFFLAG=1THENFLAG=0:GOTO3550
3510 PRINTHOME$;"[22CRSRD]";
3520 PRINT"[SHIFTP]ULSA[SPC][SHIFTR]
[SHIFTE][SHIFTT][SHIFTU][SHIFTR][SHIFTN]
[SPC]O[SPC][SHIFTQ][SPC]PARA[SPC]CAN
CELAR",
```

```
3530 INPUTANSWER$
3540 IFANSWER$="Q"THENRETURN
3550 PRINTCLSCREEN$,"[RVSON][SPC][SHIFTF]
ECHA[3SPC][SHIFTC]A[SPC]CHEQ[SPC]I
MPORTE[4SPC][SHIFTD]EPOSITO[RVSOFF]"

3560 LCNT=24
3570 RETURN
3600 IFFLAG=1THENFLAG=0:GOTO3610
3605 PRINTCHR$(12):REM SALTAR AL PRI
NCIPIO DE PAGINA
3610 PRINTTAB(26);"[4SPC][SHIFTC][SPC]
[SHIFTO][SPC][SHIFTN][SPC][SHIFTT][SPC]
[SHIFTA][SPC][SHIFTB][SPC][SHIFTI][SPC]
[SHIFTL][SPC][SHIFTI][SPC][SHIFTD][SPC]
[SHIFTA][SPC][SHIFTD]"
3620 PRINTTAB(2),"[SHIFTF][SHIFTE][SHIFTC]
[SHIFTH][SHIFTA][3SPC][SHIFTC][SHIFTA]
[SHIFTT][2SPC][SHIFTC][SHIFTH][SHIFTE]
[SHIFTQ][2SPC][SHIFTD][SHIFTE][SHIFTS]
[SHIFTC][SHIFTR][SHIFTI][SHIFTP][SHIFTC]
[SHIFTI][SHIFTO][SHIFTN][17SPC][SHIFTI]
[SHIFTM][SHIFTP][SHIFTO][SHIFTR][SHIFTT]
[SHIFTE][SPC]";
3625 PRINT"[SPC][SHIFTD][SHIFTE][SHIFTP]
[SHIFTO][SHIFTS][SHIFTI][SHIFTT][SHIFTO]
[3SPC][SHIFTB][SHIFTA][SHIFTL][SHIFTA]
[SHIFTN][SHIFTC][SHIFTE]"
3630 PRINTEQUAL$
3640 PRINT"[SPC]"
3650 LCNT=55
3660 ANSWER$=""
3670 RETURN
3800 NB$=STR$(INT(NBAL*100+.005)*.01
)
3810 FORX=1TOLEN(NB$)
3820 IFMID$(NB$,X,1)="."THENP=X:GOTO
3850
3830 NEXTX
3840 NB$=NB$+".00":GOTO3860
3850 IFP=LEN(NB$)-1THENNB$=NB$+"0"
3860 NB$=RIGHT$("[10SPC]"+NB$,9)
3880 RETURN
4000 PRINTCLSCREEN$
4020 PRINT"[8CRSRD][9SPC][SHIFTS]ALV
ANDO,[SPC][SHIFTE]SPERE"
4030 CLOSE5:OPEN5,8,5,"@0:"+CFILE$+"
,S,W"
4040 FORN=1TOLINES
4050 PRINT#5,DT$(N),CR$,BC$,BL$(N),C
R$,DS$(N),CR$,CK$(N),CR$,CA$(N),CR$,
DA$(N)
4060 NEXTN
4100 CLOSE5
4120 RETURN
4300 PRINTCLSCREEN$,"[3CRSRD][2SPC][SHIFTC]
OMPUTANDO...[SPC][SHIFTE]SPERE."
```

¡INCREIBLES TARJETAS PARA COMMODORES 64 Y VIC-20 DE 80 COLUMNAS!

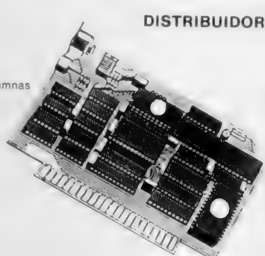
LA TARJETA DE 64K ES LA PERFECTA COMBINACION PARA LA TARJETA DE 40/80 COLUMNAS:

- Puede escoger entre 22/40/80 columnas y obtiene 31743 bytes libres!!!

- El sofware del EPROM de la tarjeta le permite utilizar todos los comandos de ficheros (OPEN, CLOSE, PRINT, INPUT, GET SAVE and LOAD) con los 31232 Bytes de Memoria «oculta», lo que produce la sensación de qué hay 8 cassettes super rápidos conectados a su Vic-20 Imagínese... menos de 1 segundo para guardar 28 K!!!

DISTRIBUIDORES PARA TODA ESPAÑA

- Tarjeta VIC-20 40/80 columnas
- Tarjeta COMMODORE 64 80 columnas
- Tarjeta VIC-20 Ampliación de memoria 64 K - RAM
- Slots de expansión para VIC-20 y 64 de 2 y 5 para conexión de tarjetas



PEDIDOS

FERRE-MORET S.A.

DISTRIBUIDORES Y PARTICULARES

C/Tusset, 8 - ENTRESUELO 2ª
Teléf.: 218 02 93
08006 BARCELONA

¡ATENCION!
SOFTWARE 80 COLUMNAS

Estos programas requieren las tarjetas distribuidas por FERRE-MORET S.A. Para el VIC-20, aunque los programas funcionan con un mínimo de 16K, están diseñados para su utilización con la Tarjeta de 64K RAM. Capacidad de almacenaje ilimitada. Todos los programas se venden con un manual explicativo en castellano

AGENDA DOMESTICA Y DE NEGOCIOS (REF. ADD4080C)

Incluye una función de búsqueda que identifica la totalidad o parte de los nombres, direcciones o números de teléfono, es decir, el sistema permite identificar un nombre, dirección o teléfono, aunque únicamente se recuerde parte de la información. Si el usuario no recuerda una dirección concreta, llamando el nombre de la población, obtendrá todas las direcciones de esa población

CONTROL STOCK 40 COL. (REF. st40C) CONTROL STOCK COL. (REF. st80C)

Posee una función de búsqueda que permite visualizar productos concretos, especificando el producto, su descripción, lugar de almacenaje, proveedor, dirección del proveedor, código de identificación del producto, precio de coste, precio de venta, cantidad (en paquetes, cajas etc.), valoración del stock, niveles máximos y mínimos de stock. Si además se dispone de impresora, se pueden obtener listados de productos específicos y listados de todo el almacén

PEDIDOS, ALBARANES, FACTURAS (REF. IDO4080CP)

Se precisa la tarjeta de 64K RAM impresora. El programa permite imprimir encabezamiento con el nombre de la Empresa y su dirección y se puede elegir que el documento sea una Factura, Albarán o Pedido. Otros detalles del encabezamiento son número de factura, fecha, número de pedido del cliente, referencia del cliente, etc. El programa permite la introducción de varios ITE's, especifica el producto, descripción, cantidad, precio antes de impuestos, precios con impuestos incluidos, gastos de envío etc. Numeración automática de los documentos. Permite hacer copias

DIETARIO DOMESTICO Y DE NEGOCIOS (REF. HBD80CDP)

Funciona con Cassette y Unidad de Disco. El programa pide la fecha e identifica cualquier entrevista concertada para ese día, detallando con quién es la entrevista, motivo de la misma, hora y lugar de la reunión, duración de la reunión, etc.
Permite la búsqueda diaria o mensual para identificar reuniones por la fecha, o lugar de reunión, o persona, tanto en el pasado como en el futuro, y además visualiza un calendario

CONTROL DE ALMACEN A 80 COL. (REF. ST80CDP)

Los datos se pueden almacenar en cassette o Unidad de Disco. Se incluye la descripción de los productos, su referencia o numero, cantidad precio de compra, precio de venta, niveles máximo y mínimo de existencias, pedidos mínimos exigidos por los proveedores, nombre y dirección del proveedor etc. Se dispone de una función de busqueda, que permite visualizar todo el almacén, productos cuya cantidad exceda la máxima autorizada, productos con existencia por debajo del nivel mínimo requerido, salidas de almacén, valor del stock, precios y márgenes

FICHERO DE PERSONAL A 80 COL. (REF. PRF80CDP)

Se necesita impresora. El programa incluye el nombre, dirección, teléfono, edad, fecha de nacimiento, numero, sexo, estado, número de Seguridad Social, Cualificaciones, Historial de empleo, Estudios, sueldo, funciones que realiza actualmente, programas de training a los que ha asistido etc. Se pueden visualizar todos los datos o seleccionar los que se desean mediante la utilización de una función de busqueda. Excelente para selección de personal, ya que permite identificación por nombre pero también se puede buscar por una cualificación o caracteristica concreta, por la edad, experiencia etc

a ACCUM, si lo tiene. De esta forma tu análisis contiene solamente números positivos, que serán más fáciles de comprender. La mayoría de la gente se confunde con números negativos.

Líneas 4620-4660.—Representan la cantidad total de los talones y la categoría del presupuesto. La subrutina en la línea 3800 es llamada varias veces para editar los números para su representación.

Líneas 4670-4680.—Calculan el porcentaje que representa la categoría del presupuesto comparado con la cantidad total de todos los talones extendidos durante el mes, representándolo en la línea 4700.

Líneas 5000-5350.—Esta rutina sirve



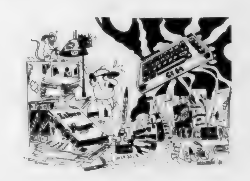
para entrar los datos que colocan el cursor en la pantalla y presenta un mensaje subrayado para que puedas ver la longitud posible de los datos. La variable LN representa el número de línea; CO representa el número de columna; SZ representa el tamaño de los datos. La línea 5180 acepta los datos en la variable E$. Las líneas 5181 a 5185 borran los subrayados de E$ después de que este haya sido entrado. Las líneas 5190 a 5330 borran el espacio donde antes se encontraban los mensajes subrayados; a continuación los datos recién entrados se vuelven a representar aquí.

Líneas 9900-9920.—Esta es la salida del programa. Se borra la pantalla y vuelve a representar las letras en mayúsculas.

◆◆◆◆◆◆◆◆◆◆◆◆

```
4310 NBAL=0
4320 FORN=1TOLINES
4330 NBAL=NBAL-VAL(CA$(N))+VAL(DA$(N
))
4340 NEXTN
4350 PRINTCLSCREEN$;"[3CRSRD][2SPC][SHIFTE]
L[SPC]NUEVO[SPC][SHIFTB]ALANCE[SPC]E
S[SPC]...";NBAL
4360 PRINT"[8CRSRD][2SPC][SHIFTP]ULS
A[SPC][SHIFTR][SHIFTE][SHIFTT][SHIFTU]
[SHIFTR][SHIFTN][SPC]PARA[SPC]SEGUIR
"
4370 PRINT"[2SPC]";:INPUTQ$
4380 RETURN
4500 PRINTCLSCREEN$;"[5CRSRD]";TAB(1
2);"[RVSON][SHIFTA]NALISIS[SPC]DEL[SPC]
PRESUPUESTO"
4510 PRINTTAB(12);"-----------------
----":PRINT
4520 PRINTTAB(7);"[SHIFTI]NTRODUZCA[SPC]
CATEGORIA";
4530 E$="END":INPUTE$
4540 IFE$="END"THENRETURN
4550 NBAL=0:ACCUM=0
4560 PRINT"[3CRSRD]";TAB(9);"[SHIFTE]
SPERE[SPC]POR[SPC]FAVOR"
4570 FORN=1TOLINES
4580 NBAL=NBAL+VAL(CA$(N))
4590 IFBC$(N)=E$THENACCUM=ACCUM-VAL(
CA$(N))+VAL(DA$(N))
4600 NEXTN
4610 ACCUM=ABS(ACCUM)
4620 GOSUB3800
4625 PRINTCLSCREEN$;"[RVSON][SHIFTA]
NALISIS[SPC]DEL[SPC]PRESUPUESTO[SPC]
PARA[SPC]";E$
4630 PRINT"[2CRSRD][SHIFTT]OTAL[SPC]
DE[SPC][SHIFTC]HEQUES...";NB$
4640 SV=NBAL:NBAL=ACCUM:GOSUB3800
4650 PRINT"[SHIFTI]MPORTE[SPC]DE[SPC]
LA[SPC]CATEGORIA..";NB$
4660 NBAL=SV
4670 IFNBAL=0THENPCT=0:GOTO4700
4680 PCT=INT(ACCUM/NBAL*100+.5)
4700 PRINT"[CRSRD]%[SPC]SOBRE[SPC]TO
DOS[SPC]LOS[SPC][SHIFTC]HEQUES.";PCT
;"%"
4720 PRINT"[14CRSRD][SHIFTP]ULSE[SPC]
[SHIFTR][SHIFTE][SHIFTT][SHIFTU][SHIFTR]
[SHIFTN]";:INPUTANSWER$
4730 GOTO4500
5000 PRINT"[HOM]";
5020 IF LN=1 THEN5050
5030 FORN=2TOLN:PRINT"[CRSRD]";:NEXT
N
5050 IFCO=1THEN5100
```

```
5060 FORN=2TOCO:PRINT"[CRSRR]";:NEXT
N
5100 PRINT"?[SPC]";
5120 FORN=1TOSZ:PRINT"[COMM@]";:NEXT
N
5125 PRINT"[HOM]";
5130 IFLN=1THEN5150
5140 FORN=2TOLN:PRINT"[CRSRD]";:NEXT
N
5150 IFCO=1THEN5180
5160 FORN=2TOCO:PRINT"[CRSRR]";:NEXT
N
5180 INPUTE$
5181 I=LEN(E$)
5182 FORN=ITO2STEP-1
5183 IFMID$(E$,N,1)<>"[COMM@]"THENN=
0:GOTO5185
5184 E$=LEFT$(E$,N-1)
5185 NEXTN
5190 PRINT"[HOM]";
5200 IFLN=1THEN5220
5210 FORN=2TOLN:PRINT"[CRSRD]";:NEXT
N
5220 IFCO=1THEN5250
5230 FORN=2TOCO:PRINT"[CRSRR]";:NEXT
N
5250 PRINT"[2CRSRR]";
5260 PRINTLEFT$(SPACE$,SZ);
5270 PRINT"[HOM]";
5280 IFLN=1THEN5300
5290 FORN=2TOLN:PRINT"[CRSRD]";:NEXT
N
5300 IFCO=1THEN5330
5310 FORN=2TOCO:PRINT"[CRSRR]";:NEXT
N
5330 PRINT"[2CRSRR]";E$
5350 RETURN
9900 PRINTCLSCREEN$;UPPERCASE$
9920 END
```



brother® EP-44

"Quien hace cosas buenas, hace buenos amigos"

La primera portátil electrónica del mundo con capacidad de TELECOMUNICACION

RECEPCION



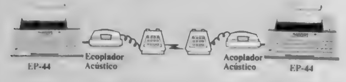
Conectada a una Computadora Personal actúa como impresora.

TRANSMISION



Transmite datos que pueden ser impresos en otras unidades.

ACOPLADOR ACUSTICO



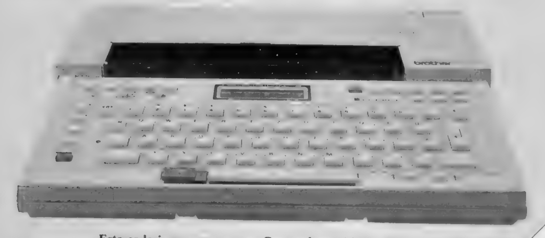
Conectada a un acoplador acústico se convierte en un completo terminal de comunicaciones.

- Terminal Input/Ouput.
- CONECTABLE A ORDENADOR Interface RS 232C. (incluido)
- Conectable a otros equipos (impresoras, máquinas de escribir, etc.).
- Teleimpresión por acoplador acústico.
- Pantalla.
- Calculador incorporado con impresión.
- Gran cantidad de escritura.
- Totalmente silenciosa

OPCIONES:

Cables de conexión para: COMMODORE 64/VIC-20
SINCLAIR ESPECTRUM
NEW BRAIN, etc., etc.

TENGA SU IMPRESORA Y MAQUINA DE ESCRIBIR POR EL MISMO PRECIO



Esta es la impresora para su Commodore-64/VIC-20

SIMO STAND Nº D-278
PABELLON XI

IMPORTADOR:
CIA DE EQUIPOS PARA OFICINA.
Enrique Granados, 65 - 08008 BARCELONA
Sta. Engracia, 147 - 28003 MADRID

Viaje al centro de tu Commodore (II)

Philip I. NELSON (RUN EEUU)

Traducido por Valerie SHANKS y adaptado por Alvaro IBAÑEZ

Terminamos este viaje por el interior de la memoria de tu ordenador



Continuando el viaje por el centro de tu Commodore en la segunda parte de este artículo, podrás ver el sistema operativo de la ROM, las zonas reservadas para el color, sonido y el manejo de los periféricos, echaremos un vistazo a los Pokes, Peeks y el cálculo del direccionamiento.

Si nos metemos por debajo de la página cero, saldremos por el bloque más alto de 8K (8000-byte) de la memoria (57344-65535). Aquí reside el sistema operativo del ordenador, un enorme programa en lenguaje máquina llamado la ROM del Kernal.

La ROM (Memoria de Sólo Lectura) es un programa que siempre se encuentra dentro del chip de silicona, incluso cuando apagas el ordenador. Sin embargo, la RAM es memoria inestable, lo que quiere decir que su contenido se desorganiza por completo cuando apagas el ordenador.

A no ser que estés dispuesto a aprender lenguaje máquina, esta zona quedará un poco fuera de tu alcance. No es que sea "Top Secret", pero sí resulta difícil de comprender para un principiante.

Sin embargo, hasta nosotros los "fisgones" somos capaces de reconocer un par de cosas en esta región. Si fisgoneas un poco por la 61624, por ejemplo, podrás ver el texto de varios mensajes que el ordenador te envía durante las operaciones Save/Load/Verify. Aquí, de nuevo, los literales se almacenan letra por letra de la misma forma que lo ve el usuario.

Un poco más abajo, por la 58464, se encuentra ese mensaje tan familiar que aparece cuando enciendes el ordenador. Más abajo todavía, en la parte inferior de esta zona, se encuentra más ROM de Kernal.

Pasando por debajo de la ROM del Kernal nos encontramos en un barrio muy movido (53248-57343), poblado de sprites, bichos ruidosos y otros animalitos. Podrías considerar este bloque de 4K como la zona de entrada/salida, que maneja mucho más de lo que nos podemos ocupar ahora en explicar.

En las posiciones 56320 a 56831 se encuentran dos chips CIA (adaptador complejo de interface) que controlan los joysticks, paddles, cronomotros y periféricos como la unidad de discos. Un poco más abajo (55296-56295) se encuentra un bloque de 1000 bytes reservado para la memoria de color. Del mismo modo que podemos controlar el formato de lo que entra en la memoria de pantalla introduciendo los valores en dicha zona mediante los Pokes, podemos controlar el color de cualquier carácter en pantalla utilizando los Pokes para meter los valores en la memoria de color.

Ambos bloques tienen el mismo tamaño, 1000 bytes, y el ordenador nos los compara automáticamente, posición por posición. Si avanzas un poco por esta zona, podrás ver que la memoria de color es una zona muy activa.

Debajo de la memoria de color, entraremos en la tierra de los ruidosos, posiciones 54272-55295, que contienen el sintetizador de sonido del C-64 (chip SID). En la parte inferior de la zona E/S se encuentra el chip VIC-II, que maneja los sprites y los gráficos de alta resolución (53248-54271). Algunas de estas zonas (como el chip de sonido) están tranquilas ahora mismo, ya que el programa "Fisgón" no los necesita. Pero da igual lo que tú o tu programa estéis haciendo, siempre hay mucho movimiento en este barrio.

Zona Aislada de RAM

Si te apetece descansar un poco, el siguiente bloque de 4K te dará la oportunidad de hacerlo. Pasa por debajo de la 53248, y encontrarás una zona que parece tener cierto parentesco con la RAM libre por donde pasamos al principio de nuestro viaje. La verdad es que eso es exactamente lo que hemos encontrado, 4000 bytes de RAM vacía, desponible para tu uso.

Pero curiosamente, este enorme trozo de memoria se encuentra aislada del resto de la RAM del usuario. Si te da la vena y escribes un programa que ocupa todos los 38K+ del espacio normal para textos en Basic del C-64, y le añades otra línea, el ordenador te enviará un mensaje de Out Of Memory Error, y dejará esta zona tan vacia como se encuentra ahora mismo.

Entonces, ¿por qué incorporar una zona aislada de RAM en un ordenador? Verás uno de los motivos si te desplazas hacia abajo hasta la posición más baja de este bloque, la posición 49152. ¿Te suena?



Tú mismo lo metiste cuando tecleaste el programa "Fisgón". Aquí es donde empieza nuestra rutina de lenguaje máquina.

Evidentemente sigue habiendo mucha RAM libre en nuestro espacio para el texto en Basic, de modo que podríamos haber metido allí el programa en lenguaje máquina (igual que hicimos con las instrucciones en pantalla), pero ¿qué hubiera ocurrido en caso de haber metido el programa en lenguaje máquina por error en medio del texto en Basic, o justo al lado suyo en los espacios que necesita para almacenar las variables o los "arrays"? El resultado hubiera sido un desastre soberano. Para que la vida sea más sencilla, guardamos la rutina en lenguaje máquina aquí arriba, donde el Basic nunca se meterá.

Para que puedas leer el programa en lenguaje máquina que tienes delante, utiliza la tabla de códigos de pantalla. Durante la iniciación, el primer valor de datos que introdujimos mediante en Poke fue 160, que aparece como un cuadro blanco (espacio inverso) en la pantalla.

Ya que estamos aquí, observa lo compacto que resulta el programa en lenguaje máquina, comparado con nuestro texto en Basic. Ocupa solamente 69 bytes, pero no tarda nada en hacer lo que el Basic llevaría muchos segundos aburridos en realizar. Los programas en lenguaje máquina son tan cortos porque están escritos en la lengua materna del ordenador: los números: La mayoría de los caracteres que ves aquí representan instrucciones, parecidas a una palabra en Basic, que el ordenador ejecuta al instante. No se tiene que preocupar de los números de línea, ni los punteros, trabaja con valores limpios y ligeros.

El ordenador realiza el mismo proceso de verificación de errores mientras está ejecutando tu texto en Basic.

Otro Programa en Basic

Sigue bajando hasta la siguiente zona de 8K, y descubrirás más explicaciones para esta gran diferencia de velocidad. Esta zona (los bloques de 8K son muy populares aquí en la parte alta de la memoria) tiene un alcance desde la 40960 hasta la 49151. A medida que bajas, estarás pensando que todo esto se parece mucho al programa del sistema operativo que vimos al principio de la memoria. Si es asi, has vuelto a acertar. Estamos viendo otro enorme programa en lenguaje máquina, el que la mayoría de nosotros llamamos Basic.

A lo mejor nunca lo has considerado de esta forma, pero el lenguaje Basic es simplemente otro programa, registrado permanentemente en la ROM. La finalidad del programa Basic es la de ayudarte a ti a escribir programas. Si sigues sin creértelo, baja a la 40960 donde empieza el Basic. Igual que "Fisgón", este programa tiene un título: CBM Basic. Más abajo todavía las palabras clave y los mensajes de error del Basic están almacenados como literales. Vamos a parar un momento a reflexionar —aquí hay mucho que aprender.

Este programa en Basic (normalmente denominado Intérprete Basic) que realiza muchos trabajos y detecta muchos de tus errores, es lo que hace que tu ordenador sea una máquina tan amistosa. Cada vez que enciendes el ordenador, el sistema operativo lo hace pasar por una serie de rutinas de inicialización (una de las cuales, como hemos visto, es fijar el puntero que indica el principio del Basic). En cuanto el ordenador termine sus tareas domésticas, vuelve automáticamente al programa en Basic y lo empieza a ejecutar.

Vamos a suponer que acabas de encender el ordenador. El Basic se está ocupando de sus cosas tranquilamente en la trastienda. Aunque el ordenador solamente "entiende" números, se deja mandar por unas palabras estilo inglés. Si tecleas la primera línea de "Fisgón" y pulsas la tecla Return, el ordenador almacenará su versión abreviada al principio del espacio reservado para el texto en Basic. Teclea la segunda línea, y el texto irá creciendo hacia arriba en la memoria. Además de comprimir, numerar y almacenar las líneas de tu programa y manejar sus punteros, el ordenador está comparando tu sintaxis con el diccionario que forma parte del Basic.

Por ejemplo, si en vez de entrar una línea numerada en Basic, introduces la palabra TONTERIA, el ordenador te enviará un mensaje de error. Tu entrada no lleva número de línea, de modo que no puede ser almacenada como parte del texto en Basic, y el ordenador ha comprobado tu palabra en su diccionario incorporado y ha descubierto que TONTERIA no existe en el vocabulario de CBM Basic. Esto significa que el ordenador tampoco puede tratar tu palabra como un comando directo en el modo inmediato. Sólo te puede avisar que existe un error de sintaxis, y que lo intentes de nuevo.

El ordenador realiza el mismo proceso de verificación de errores mientras está ejecutando tu texto en Basic. A medida que va leyendo, el ordenador reconoce cada palabra clave en Basic comparándola con las que tiene en su lista de palabras. Si encuentra una palabra igual, sigue adelante. Lo que ocurre a continuación depende de varias cosas, de modo que vamos a considerarlas una por una.

Vamos a suponer que tu texto en Basic contiene la sentencia POKE A, B, y que el ordenador acaba de comprobar que Poke es una palabra en Basic. A continuación tiene que verificar lo que representa A, para lo cual necesita desplazarse a la casilla A en el espacio para el almacenamiento de las variables. Ahora el ordenador sabe la posición donde queremos realizar el Poke, todavía no sabe el valor que le queremos introducir. De modo que tiene que ir a la casilla que contiene la B, para recoger el valor de B para poder realizar nuestro comando, es decir, en cuanto termine de hacer unas cuantas verificaciones.

¿Qué ocurriría si le dijéramos al ordenador que realizara un Poke en una posición que no exisitía, o en una posición mayor que la 65535? O, ¿qué pasaría si la posición es la correcta, pero le hemos obligado al ordenador a introducir un valor demasiado grande para cualquier celda de memoria? Pues, el ordenador reconocerá el error, sacará el mensaje de Illegal Quantity Error de la ROM y lo pondrá en la pantalla y hará parar el programa. Además, no dirá cuál es la línea que contiene el error.

Pero no saques demasiadas conclusiones. Solamente porque el ordenador dejó de ejecutar nuestro texto en Basic no significa que ha dejado de funcionar en Basic. Lo único que ha hecho ha sido dejar el modo de ejecución para adoptar el modo de edición para que podamos arreglar nuestro texto. Esta parte del programa en Basic nos permite listar la línea incorrecta, modificarla y sustituirla por la correcta. Después de haber almacenado la línea nueva, el ordenador ajustará todos sus punteros de acuerdo con sus necesidades, y esperar que el humano le dé un comando que tenga un poco de sentido.

Por lo tanto, mientras que tú estás haciendo tu programa, el ordenador está



ejecutando el programa en Basic al mismo tiempo, ocupándose de que nuestro camino sea más fácil de seguir. Este proceso tan complejo de traducir nuestros comandos y captar nuestros errores es lo que hace que los programas escritos en Basic vayan tan despacio. Cuando exigimos que nuestro ordenador realice todo esto a la velocidad de la luz, le estamos pidiendo mucho; además, tiene otras cosas que hacer, como por ejemplo enviar una imagen en vídeo que tu pantalla puede volver a sacar 60 veces cada segundo.

Casi hemos llegado

Si has aguantado hasta ahora, espérate un poco más. Estamos a punto de entrar en tierras más familiares —el RAM libre utilizada para el texto en Basic. En el C-64, ésta llega desde la 2048 hasta la 40959. Como verás ahora mismo, esas son muchas casillas. Pero antes de reemprender el largo camino hacia la representación de nuestras instrucciones, vamos a mirar esa cuarta zona "extra" reservada por el ordenador cuando se ejecuta un programa.

Esta zona se utiliza para almacenar los "strings". A diferencia del espacio de

Aunque no vamos a tocar los motivos ahora, esto te permite realizar grandes cosas con el Basic

almacenamiento de las variables y los "arrays", que empiezan desde abajo y crecen hasta arriba, el almacenamiento de los "strings" empieza en la parte superior de la RAM del usuario y crece hacia abajo. De hecho, este espacio se ha ido aumentando a lo largo de nuestro viaje. No te puedo decir exactamente dónde se encuentra el borde inicial del almacenamiento de los "strings" porque no sé el número de veces que has pulsado una tecla mientras has estado fisgoneando. Se encuentra en algún lugar debajo de la 40959, y se desplaza más abajo cada vez que pulsas una tecla.

Desplaza el cursor hacia abajo poco a poco y encontrarás el almacenamiento de los "strings" representado como una línea de caracteres "q" que se desplazan hacia la izquierda de la pantalla. Si no estás muy seguro si lo has encontrado, mantén la barra de espacios pulsada y verás un bloque oscuro y sólido que se desplaza de derecha a izquierda, desde las posiciones más altas hasta las más bajas.

Lo que estamos haciendo es pasar una y otra vez por las sentencias GET A$ en la línea 4. Si has pulsado una tecla distinta cuando el ordenador llega a esta sentencia, la variable llamada A$ se vuelve a definir, y el "string" nuevo se almacena por separado aquí arriba. Aunque no vamos a tocar los motivos ahora, esto te permite realizar grandes cosas con el Basic. En el programa "Fisgón", cada "string" nuevo tiene una longitud de un carácter, pero el CBM Basic te permite hacer "strings" con una longitud de hasta 255 caracteres juntando (concatenación) unos "strings" cortos.

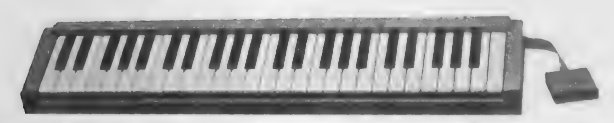
Si usas cientos de sentencias Get para introducir datos mediante el teclado esta zona resulta fácil de encontrar, pero esta técnica entraña sus peligros. Mientras escribo esto, el principio del espacio para los "strings" se encuentra por la 38000 en la memoria, es decir muy lejos de nuestro texto en Basic, aunque la pantalla de instrucciones se encuentra mucho más alta (12288-12387). Si el almacenamiento de los "strings" llega hasta allí abajo, pasará por las posiciones de la pantalla de instrucciones, llenándolas de unas "q" sin sentido. Ten en cuenta que existen más métodos para traer los datos introducidos mediante el teclado (haciendo un Peek en la Posición 197, por ejemplo) que no comen la memoria de esta forma.

Estamos llegando a casa, de modo que vamos a recorrer el largo camino hasta la pantalla de instrucciones. No me pude hacer idea de la cantidad de RAM para el usuario que contiene el C-64 hasta que realicé este viaje desde la parte superior. No te preocupes si tus instrucciones están todas mezcladas, el formato de antes sigue estando allí, almacenado en exactamente las mismas posiciones. Durante nuestro paseo por el C-64, hemos ido dando pasos de 40 bytes para que la imagen en pantalla tuviera alguna coherencia. El número 40 no entra en 65536, que es el número de posiciones de memoria por las cuales acabamos de pasar, de

MUSIC 64

PERSONAL
COMPUTER
MUSIC





MUSICA CON COMPUTADOR con el Teclado «MUSIC 64»

El teclado «MUSIC 64» se empalma directamente al punto de intersección del computador del COMODORO 64. Su diseño armoniza perfectamente con el del computador. La buena calidad de los materiales permiten el uso del teclado.

Se puede emplear como sintetizador monofónico o bien como teclado de órgano polifónico. Para éllo se necesita lo siguiente:

- un COMODORO 64
- un video monitor o un receptor de televisión
- y una unidad FLOPPY DISK o de cassettes

ALBAREDA le proporciona a Ud.

- un teclado de 4 octavas de Do a Do
- un adaptador para realizar el empalme del teclado con el computador
- los materiales de calidad excelente de MUSIC 64

PREPARACION

Conecte el COMODORO 64 al monitor video y al FLOPPY DISK o de cassettes (consultando el manual para el uso del COMODORO). No lo enchufe todavía introduzca el conectador del teclado «MUSIC 64» dentro del punto de intersección, de manera que el cable salga hacia atrás (los contactos situados en la parte de delante están destinados al computador) meter el estuche de compases (adaptador) dentro del refugio de expansión del COMODORO. Atenerse a la indicación «up» «hacia arriba» y «down» «hacia abajo».

Coloque todas las piezas preparadas para su empleo.

PARA USAR EL MUSIC 64

Enchufe el COMODORO 64, el video monitor y el FLOPPY DISK, entonces aparecerá en el video monitor la indicación COMODORO BASIC.

UTILIZACION DE DISCOS

Colocar el disco, cara arriba en el FLOPPY DISK y cerrarlo
use el teclado del computador para escribir LOAD «MONO 64» (para el sintetizador monofónico) o bien LOAD «POLY 64», 8 (para el teclado polifónico)
apriete el pulsador que dice RETURN (Retorno)
el computador se hace cargo del programa del disco y anuncia que está preparado con la palabra «READY» (Preparado)
ahora soltar el pulsador RUN (en marcha) y apretar el pulsador «RETURN» (Regreso)
las demás instrucciones para su utilización se consiguen por medio del diálogo directo con el video monitor

UTILIZACION DE CASSETTES

Colocar la cassette «MUSIC 64» en su aparato de funcionamiento, si fuera necesario, mover la bobina hacia atrás
con la ayuda del teclado de la computadora escriba LOAD «MONO 64» (para el sintetizador monofónico) o bien
LOAD «POLY 64» (para el teclado polifónico)
apriete el pulsador que dice «RETURN» (Retorno)
cumpla la orden que le indica el video monitor «PLAY PRESS ON TAPE», pulsando la tecla PLAY del indicador
El computador se hace cargo del programa de la cassette y anuncia «READY» «PREPARADO»
ahora soltar el pulsador RUN (en marcha) y apretar el pulsador RETURN (Retorno)
las demás instrucciones para su utilización las conseguirá por medio del diálogo directo con el video monitor

Advertencia: los márgenes de tolerancia de los materiales empleados en la fabricación (en el ámbito del audio del computador y del monitor) pueden hacer alterar el sonido de un computador a otro computador. Si fuera preciso cambie adecuadamente los parámetros del filtro

El sintetizador monofónico

La relación de los instrumentos de los cuales se puede disponer está indicada en el video monitor y son los siguientes:

0	Trompeta	7	Instrumentos de cuerda (cuerdas)
1	Brass	8	Piano
2	Clarinete	9	Organo eléctrico I
3	Campanas	A	Organo eléctrico II
4	Flauta	B	Acordeón
5	Guitarra	C	Random (sintetizador)
6	Wha Brass (cobres-wha)		

Para elegir o cambiar un instrumento, basta con apretar el pulsador adecuado del computador correspondiente al código (del 0 al C). El pulsador « ↑ » suelta el instrumento elegido para que toque

Para modificar las características paramétricas del instrumento utilice los pulsadores que se indican a continuación:

F1 para alcanzar el parámetro siguiente
F3 para regresar al parámetro precedente
F5 para aumentar el valor del parámetro elegido
F7 para disminuir el valor del parámetro elegido

Mientras se realizan tales cambios, el video monitor indica «MODIFY MODE» «MANERA DE MODIFICAR» en vez del nombre del instrumento

Para volver a obtener los valores iniciales de los parámetros de un instrumento (después de haber hecho varios cambios), hay que pulsar en el teclado del COMODORO 64, la tecla «CLR Home»

El teclado del órgano polifónico

La relación de los instrumentos de los cuales se puede disponer está indicada en el video monitor y son los siguientes:

1 Spinete	4 Flauta
2 Acordeón	5 Banjo (especie de guitarra de los negros)
3 Campanas tubulares (tecnica del instrumento con el timbre)	

Para elegir o cambiar un instrumento basta con apretar el pulsador (1 al 5) del computador, correspondiente al código. El pulsador ↑ sirve para soltar el instrumento elegido, para que toque. El pulsador M le permite interrogar de nuevo la relación de instrumentos

Para cambiar los parámetros de uno de estos PRE SETS (preselección), pulsar una de las teclas comprendidas desde F1 a F8.

En este caso, en lugar de aparecer el nombre del instrumento, aparece la indicación: «MODIFY MODE» «MANERA DE MODIFICAR»

Para cambiar los parámetros utilizar los pulsadores de la siguiente forma:

F1	Ataque	F2	Volumen
F3	Declive	F4	Traslado
F5	Sostener	F6	Forma de la onda
F7	Soltar	F8	Ciclo de la operación

El valor del parámetro puede aumentarse apretando repetidas veces el pulsador correspondiente, después de haber alcanzado el valor máximo, se elige el valor mínimo

Para obtener el valor de origen, pulsar la tecla «R» (Reset) (Reengarzar)



INSTRUMENTOS
Y ACCESORIOS MUSICALES

C/. Carmen, 19
TARREGA (Lérida) - Teléfonos (973) 31 04 02 - 31 23 51

modo que no hemos podido aterrizar exactamente en la base. Y además, seguramente estarás todavía en la modalidad de gráficos minúsculas. Para que todo vuelva a la normalidad, pulsa la tecla Q para salir, y vuelve a teclear RUN.

Cómo funciona "Fisgón"

Ahora que has visto las entrañas de tu ordenador, seguramente querrás saber cómo "Fisgón" las ha podido sacar en pantalla. Yo había escrito la parte de lenguaje máquina de este programa para un juego en el que el jugador puede hacer un scroll hacia arriba o hacia abajo dentro de un terreno de juego con una longitud de muchas pantallas, que yo había diseñado en la memoria igual que la pantalla de instrucciones de "Fisgón". Una vez que me di cuenta de la forma de desplazarme por el terreno de juego, supe que no tenía por qué parar allí usando la misma técnica, me podía meter en cualquier parte del ordenador, abrir la puerta y ver todas esas cosas misteriosas e invisibles que conocía solamente de los libros.

Tu ordenador dispone de dos ventanas incorporadas para mirar la memoria. Una es la sentencia Peek, que te proporciona el valor numérico almacenado en cualquier posición de memoria. La sentencia Peek te puede llevar desde cero hasta 65535, pero no deja de ser una ventana. La memoria de pantalla es otro tipo de ventana, miles de veces más grande. Cualquier cosa que se introduce mediante un Poke en esta zona, sale en la pantalla, ¿pero cuál es el método para que esta ventana enorme se mueva?

Pues bien, la memoria de pantalla tiene una movilidad muy limitada. Si haces un Poke (con cuidado) en una posición de control, 53272, podemos desplazar la memoria de pantalla hacia cualquiera de estas dieciséis posiciones en el C-64. Pero todas estas posiciones son fijas, y esta técnica crea más complicaciones. Parece imposible hacer un scroll de unos cuantos bytes hacia arriba o hacia abajo. ¿Cuál es la solución?

Si no podemos llevar la pantalla a los datos, tendremos que llevar los datos a la pantalla. Resulta fácil (pero lento) en Basic copiar cualquier bloque de memoria de mil bytes en la pantalla. Solamente hay que hacer un Peek en cada posición original de modo consecutivo, e introducir los valores que encontramos en la memoria de pantalla mediante un Poke. Vamos a hacer una pequeña prueba de velocidad. Si todavía se está ejecutando el programa "Fisgón", pulsa la F para salir. Ahora, borra la pantalla, entra la siguiente línea y pulsa la tecla Return.

```
FOR J=0 TO 999: POKE 1024+J,
  PEEK(12288+J) :NEXT J
```

¡Voila! El bloque original que hemos copiado es el que contiene nuestras instrucciones. Para hacer un scroll de una línea hacia arriba o hacia abajo en la memoria, podemos sumar o restar 40 de 12288, y repetir todo el proceso. De acuerdo, esto funciona, pero tardaríamos



muchísimo en mirar las 65536 posiciones en Basic. Nuestro programa en lenguaje máquina todavía se encuentra en la memoria, de modo que vamos a ver si nos puede acelerar un poco todo el proceso. Borra la pantalla y entra esta línea:

```
POKE 251,0: POKE 252,48: SYS 49152
```

Supongo que te habrás dado cuenta quién ha ganado la carrera. Además de copiar 1000 bytes, la rutina en lenguaje máquina acaba de llenar todas las 1000 casillas de la memoria de color con valores en blanco para las "nuevas" ROM del C-64. Podríamos decir que "Fisgón" es un programa híbrido, dado que funciona en dos partes, cada una cumpliendo la función que mejor sabe desempeñar. La rutina en lenguaje máquina realiza una sencilla tarea repetitiva de una manera enormemente eficaz, y el Basic maneja unos cálculos más complejos donde la velocidad no tiene tanta importancia.

Este tipo de programa en lenguaje máquina es conocido por una utilidad de movimiento en bloque, y, dada la memoria grande disponible en el C-64, sus posibilidades no tienen límite. Podrías establecer diferentes pantallas en varios

El ordenador puede quedarse colgado si haces un Poke a ciegas en las zonas sensibles como la página cero.

puntos a lo largo de la RAM, y hojearlas a la velocidad de la luz, para cambiar de un terreno a otro en un juego, escribir páginas consecutivas de instrucciones, etcétera. Antes de poder realizar su trabajo, la rutina en lenguaje máquina requiere solamente un dato: la dirección inicial del bloque original de 1000 bytes de memoria que queremos copiar.

Cambiando la dirección

Es la obligación del programa en Basic es saber la dirección a donde se tiene que dirigir, y este tipo de transferencia de información es una parte fundamental de la informatica, de modo que vamos a estudiarlo un poco más de cerca. Ya has aprendido cómo el ordenador almacena los valores de las variables en unas "casillas" especiales, reservadas y etiquetadas precisamente para este propósito. Siempre que se modifique una variable, o se necesita en otro sitio, el ordenador vuelve a la casilla para almacenar un valor nuevo o recuperar el valor actual. Nosotros estamos haciendo exactamente lo mismo con nuestro programa "Fisgón". Cada vez que pulsas la tecla cursor arriba/abajo, el programa calcula una nueva dirección original y la almacena en la casilla donde sabe que la buscará la rutina en lenguaje máquina.

Si resulta difícil hacerte la idea, imagina que la rutina en lenguaje máquina es un mensajero. El jefe, el programa en Basic, escribe la dirección en un papel que mete en la casilla mientras el mensajero está durmiendo. El jefe despierta al mensajero con el comando SYS. El mensajero va corriendo a la casilla asignada, lee la dirección, y se dirige hacia ella corriendo para recoger los 1000 bytes de datos y entregarlos donde se necesitan (la memoria de pantalla).

Hasta ahí resulta sencillo, pero esto se complica un poco porque el ordenador necesita dos células de memoria para almacenar la mayoría de las direcciones. De modo que nuestra casilla consiste en dos células, una al lado de la otra. Recuerda, cada célula sólo puede almacenar un número (decimal) de cero hasta 255, mientras que el ordenador dispone de direcciones que pueden llegar hasta 65535. Para que lo puedas comprender mejor, vamos a estudiar un poco el sistema de numeración binario.

Todos conocemos el sistema de numeración decimal, llamado de base 10 porque tiene 10 dígitos: de 0 hasta 9. Pero se puede contar de muchas otras formas. Un chip de silicona es mucho menos complejo que tu cerebro, de modo que resulta más fácil para los ordenadores emplear números binarios, de base 2, que solamente utilizan dos dígitos: 0 y 1. Utilizando este sistema sencillo, podemos escribir cualquier número de 0 hasta 255 empleando solamente ocho dígitos. El número decimal 1 tiene el siguiente formato: 00000001, y el número decimal 254 este: 11111110.

Imagina que cada célula de memoria es una colección de ocho interruptores que ▶

pueden estar en posición de encendido o apagado y de esta forma representar un 1 ó un 0 en binario para el ordenador. Esto funciona si nos quedamos en la página cero, donde todas las direcciones son menores que 256. Pero la dirección más alta del ordenador, 65535, ocupa dieciséis bits, y tiene el siguiente formato: 1111111111111111. Esto representa dos veces el tamaño de cualquier célula de memoria.

La solución es hacerlo por duplicado. Los microprocesadores 6502 utilizados en los ordenadores Commodore, Apple y Atari leen las direcciones mirando una pareja de números que residen en dos células de memoria contiguas, sumando las dos mitades de ocho bits para formar una dirección entera de dieciséis bits. En el programa "Fisgón", el jefe Basic tiene que meter la mitad de la dirección en una casilla (posición 251) y la otra mitad en la célula al lado (252).

Si has llegado hasta aquí, no te rindas ahora. El último problema que tiene que ser resuelto es que el mensajero es bizco, no es capaz de juntar las dos mitades de una dirección hasta que el jefe las invierte. Si volvemos a pensar en el sistema binario, esto significa que los ocho dígitos más altos (los más importantes) tienen que entrar en la segunda casilla, y los ocho dígitos más bajos (los menos importantes) entran en la primera. Repasa la línea 4 de



"Fisgón", y verás que LO se introduce mediante un Poke en la 251, y Hi se introduce en la 252. Esto se llama frecuentemente format de byte bajo/byte alto.

Al principio este sistema te podría parecer de locos, pero merece la pena aprenderlo si pretendes ir más allá de la programación más sencilla. La mayoría de los ordenadores personales utilizan el microprocesador 6502, lo que significa que el formato byte bajo/byte alto tendrá muchos años de vida. Parece complicado, pero si lo tomas con calma verás cómo al final te resulta sencillo.

De todas formas, una vez que comprendas lo que representan LO y HI, el resto del programa "Fisgón" es sencillo. La pantalla no se mueve nunca. Lo único que hacemos es cambiar la dirección del bloque original que queremos copiar, enviar la dirección byte bajo/byte alto a las casillas correspondientes y la velocidad del lenguaje máquina hace el resto.

Por supuesto, todavía no has visto las entrañas de tu ordenador; hay poca luz dentro de un chip, y de todas formas, no podrías ver los electrones. Pero este programa te da una idea, y cuanto más veas, más te apetece aprender.

Si te has cansado de "fisgonear", intenta variar un poco. Pulsa la P para llamar la rutina de los Pokes, e introduce dos valores, siendo el primero la dirección y el segundo el valor donde vas a hacer el Poke.

Si no estás muy familiarizado con los Pokes, prueba con algo sencillo: coloca un carácter gráfico en una zona en blanco de la pantalla de instrucciones. Pero ten cuidado, el ordenador puede quedarse colgado si haces un Poke a ciegas en las zonas sensibles como la página cero.

Fisgoneando en el VIC-20

El VIC-20 y el C-64 son dos máquinas muy distintas, pero en este caso los dos pueden usar casi el mismo programa. El VIC-20 dispone de mucho menos RAM del usuario, por lo que omití la pantalla de instrucciones, pero al mismo tiempo intenté subrayar el parecido entre los miembros de la familia Commodore al hacer dos listados similares.

El programa "Fisgón" para el VIC-20 es solamente para el VIC no ampliado, de modo que tendrás que quitarle cualquier cartucho RAM antes de ejecutarlo. La ampliación de la RAM traslada la memoria de pantalla, la memoria de color y el texto en Basic a otras posiciones según la cantidad que se añada. Por este motivo es aconsejable consultar la "Guía de Referencia del Programador para el VIC-20" si quieres modificar el programa para que funcione en una máquina ampliada. También tendrás que modificar la rutina en lenguaje máquina para poder enviar los datos a las nuevas posiciones de pantalla.

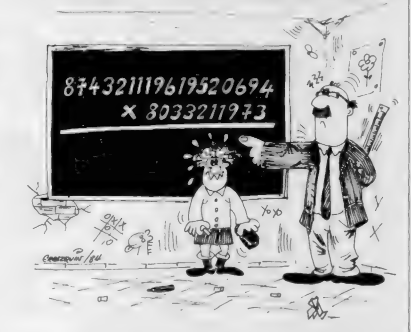
Al arrancarse, el programa "Fisgón" te coloca en la parte inferior de la zona de almacenamiento del texto en Basic, en las posiciones 4096-4601. La memoria de pantalla es más pequeña (506 bytes comparados con 1000 bytes en el C-64), lo que significa que tu viaje se hará en saltos de 22 bytes, la longitud de una línea de pantalla en el VIC. Dado que estamos copiando bloques de memoria más pequeños, la rutina en lenguaje máquina también resulta más corta. A diferencia del C-64, el VIC-20 no dispone de RAM "extra", por lo que la rutina en lenguaje máquina fue colocada en el buffer del cassette, empezando en la posición 828.

Te incontrarás con el mismo equipo básico dentro del VIC-20: el Basic CBM y la ROM del Kernal de 8K, más una página cero casi idéntica. Pero todo esto se encuentra en diferentes sitios (consulta los mapas de memoria en el manual), y además existen otras diferencias importantes entre los dos ordenadores.

Obviamente, el VIC-20 no tiene sprites, y el generador de sonido no es tan lujoso como el chip SID del C-64. Y aunque el microprocesador 6502 del VIC, igual que el 6510 del C-64, puede acceder a cualquier posición de memoria de cero hasta 65535, existen grandes "trozos" del VIC que no se utilizan para nada. Son simplemente agujeros en la memoria, esperando ser llenados de cartuchos de expansión de la RAM, de una ROM nueva, o de lo que sea.

Pero no subestimes tu VIC-20. Sus grandes caracteres programables son dos veces el tamaño de los del C-64, y además, puede hacer cosas que su hermano mayor no puede, cómo modificar el centrado de su pantalla en un monitor. Aunque dispone de poca memoria, contiene un potente microprocesador que se ajusta a las normas de la industria, y si llegas a dominar esta máquina, no tendrás ningún problema en entender cualquier otro ordenador.

Multiplicación larga en Basic



Los Commodores no siempre proporcionan la respuesta correcta para multiplicaciones realmente larga. Este programa proporciona la precisión matemática necesaria para ti y para tu Commodore.

Un amigo mío vino a casa un día diciendo que habia perdido cien pesetas apostando con un niño que decía que los equipos Commodore eran bastante buenos pero no eran ninguna maravilla. Para demostrar lo que queria decir apostó con mi amigo que el ordenador era incapaz de multiplicar dos números y dar un resultado correcto. El niño hizo que el ordenador multiplicara dos números de 10 digitos y el resultado salió como uno de esos números de punto flotante con un exponente. El niño tenia razón: el resultado no era correcto.

Esta historia provocó mi curiosidad. Efectivamente, ¿cómo se podía programar un ordenador Commodore para que multiplicara largas cadenas de números? Yo sólo me acordaba de un campo en que esta multiple precisión matemática se utilizaba: la criptografía moderna, donde se utilizan largos números de más de 70 digitos con mucha frecuencia.

Parecía un reto divertido intentar escribir un programa para solucionar este problema y, dado que no tenía el tiempo para emplear ningún método un poco sofisticado, decidí imitar la multiplicación a mano. El resultado se presenta en el Listado 1.

Sobre el programa

Se empieza determinando tres "arrays": M1 para el primer factor (multiplicando), M2 para el segundo factor (multiplicador)

.(Pasa a pág. 74)

VIC-20 - C-64
CBM-100 PETS

```
10 REM MULTIPLICACION LARGA EN BASIC
15 REM ESCRITO POR
20 REM ALEX LANE
40 REM
45 REM ESCRITO PARA
50 REM VIC-20 Y C64
60 REM ESTE PROGRAMA
65 REM ENCUENTRA EL
70 REM PRODUCTO DE DOS
75 REM NUMEROS.
80 REM TENIENDO CUALQUIERA DE LOS
85 REM DOS UNA LONGITUD MAXIMA
90 REM DE 70 DIGITOS.
95 REM
100 DIM M1(70),M2(70),A(140)
105 REM
110 REM INPUT
115 REM
180 PRINT"[CLR]" INPUT"PRIMER[SPC]NU
MERO",A$
183 IF LEFT$(A$,1)="-"THEN FL=FL+1 A
$=RIGHT$(A$,LEN(A4)-1)
184 GOSUB 600
185 FOR I=LEN(A$) TO 1 STEP -1
190 M1(I)=VAL(MID$(A$,LEN(A$)-I+1,1)
)
195 NEXT I
200 INPUT"SEGUNDO[SPC]NUMERO",B$
203 IF LEFT$(B$,1)="-"THEN FL=FL+1:B
$=RIGHT$(B$,LEN(B$9-1)
204 GOSUB 600
205 FOR I=LEN(B$) TO 1 STEP -1
210 M2(I)=VAL(MID$(B$,LEN(B$)-I+1,1)
)
215 NEXT I
217 IF A$="0"OR B$="0" THEN AN$="0"
GOTO 550
```

```
220 REM
225 REM EJECUTANDOSE
230 REM
305 REM
310 FOR J=1 TO LEN(A$)
320 FOR I=1 TO LEN(B$)
330 F=I+J-1
340 A(F)=A(F)+M2(I)*M1(J)
350 IF A(F)>9 THEN A(F)=A(F)-10 A(F+
1)=A(F+1)+1:GOTO 350
360 NEXT I
370 NEXT J
380 REM
385 REM OUTPUT
390 REM
500 FOR I=140 TO 1 STEP -1
510 IF LEN(AN$)=0 AND A(I)=0 THEN GO
TO 530
520 AN$=AN$+RIGHT$(STR$(A(I)),1)
530 NEXT I
540 IF FL=1 THEN AN$="-"+AN$
550 PRINT PRINT"EL[SPC]PRODUCTO[SPC]
ES "
560 PRINT AN$
565 END
600 REM
610 REM VERIFICACION
620 REM
630 IF F1=0 THEN C$=A$
640 IF F1=1 THEN C$=B$
650 FOR K=1 TO LEN(C$)
660 IFVAL(MID$(C$,K,1))=0THENIFMID$(
C$,K,1)<>"0"THENPRINT"MAL[SPC]DATO"
GOTO565
670 NEXT K
680 F1=1
690 RETURN
```

MusiCalc™

El Sistema Creativo de Música
(para personas aún sin conocimientos musicales)

¡Que suene la música!
Convierta a su Commodore 64 en un sofisticado instrumento musical, y él le convertirá en un compositor, director y músico.

Con **MUSICALC**, el sistema de música creativa, toda la familia es capaz de componer y tocar música de una manera instantánea.

¡**MUSICALC** le divierte jugando con la música!

Obtenga copias en papel de sus propias partituras, utilize más de 30 escalas distintas, conéctelo a su equipo de alta fidelidad y a una caja de ritmos, y muchas, muchas posibilidades más.



TABLA GRÁFICA
para Commodore 64



Entra en el apasionante mundo del arte de ordenador.
¡Fácil de usar!

Se entrega junto con soft en cartucho. Gráficos en alta resolución. 16 colores. 2 páginas de dibujo. Zoom, menús de diseño, archivo en cinta o disco, posibilidad de incluir texto. Permite printar los gráficos mediante soft adicional.



Mando de juego

Durabilidad y buen precio

Compatible con C-64, Atari, Spectravideo, Sinclair Spectrum (con interface) etc...

MONITOR DE COLOR
14" Profesional



- Pantalla de gran resolución
- Sonido incorporado
- Entrada señal video compuesta o RGB
- Peana orientable
- Gran resistencia y durabilidad

Adaptable a todos los ordenadores.
(Los microordenadores que sólo tienen salida RF son compatibles mediante una sencilla conexión)

Distribuidor exclusivo



Gran Vía Carlos III, 97 K. 08028 BARCELONA
Tels: 330 33 08 - 330 33 62. Télex: 50156 JONC

Le presentamos a SPINNAKER.
Para que también nuestros hijos aprendan jugando con el ordenador.

Muchos padres ya conocen que el ordenador doméstico puede servir para que nuestros hijos jueguen y pasen largos ratos de ocio. Con SPINNAKER los **juegos de ordenador** además sirven **para aprender**.

SPINNAKER es la firma líder mundial en juegos educativos para ordenador. Ahora llega a nuestro país para que también nuestros hijos aprendan jugando con el ordenador.

Nuestros juegos están llenos de emoción, entretenimiento, color, música... Pero además poseen un alto grado de valor educativo, ayudando a los niños a que desarrollen sus habilidades y aprendan de una divertida manera. Aprovechando al máximo las nuevas posibilidades que nos ofrece el mundo de las computadoras.

En su elaboración han intervenido especialistas en la educación, la didáctica, la música, los gráficos y la programación, que han realizado un cuidado producto especialmente pensado para la educación.

Estos son los motivos por los que miles de niños de todo el mundo son verdaderos fans de SPINNAKER. Y sus padres también.

Si Ud. realmente desea que su ordenador familiar sea una ayuda útil para el futuro de sus hijos, recuerde este nombre: SPINNAKER: Una divertida familia de juegos educativos.



DISPONEMOS DE VERSIONES PARA APPLE II, Commodore 64 y Sinclair Spectrum

Precio especial de introducción 2.400 pts.



Gran Vía Carlos III, 97 K. 08028 BARCELONA



SPINNAKER SOFT COR. CAMBRIDGE, MASS. USA

Rutinas Basic (1)

Formatación numérica

Jordi Sastre

Abrimos aquí una nueva serie de artículos dedicada a las rutinas Basic. Normalmente todo programador necesita construirse rutinas para formatar datos numéricos (a menos que se disponga de PRINT USING), efectuar entradas de datos controlados (rutinas sustitutivas del comando INPUT), paginar los listados con cabeceras y numeración de páginas, etc.

Lógicamente cada programador tiene sus gustos y manera de trabajar. Es difícil que existan dos rutinas iguales hechas por programadores diferentes. Ambas podrán cumplir exactamente la misma función, pero siempre habrán diferencias en cuanto a presentación, rapidez de ejecución, eficacia, etc.

Las rutinas cambian de manos: un programador se las pasa a otro, se publican en revistas, se extraen de programas ajenos, etc. Cuando una rutina llega a mis manos, lo primero que intento hacer es mejorarla, cosa unas veces posible y otras no. Lo que es más factible es modificar para adaptarla a una filosofía de trabajo más particular. Pretendo que el lector haga lo mismo: no limitarse a utilizar la rutina sino intentar mejorarla u optimizarla. Espero dentro de pocos meses ver en esta revista una nueva sección titulada: Rutinas Basic Mejoradas.

Formatación numérica

Vamos pues a atacar la primera de las rutinas, la FORMATACION NUMERICA. Para imprimir la variable numérica A basta con efectuar PRINT A. Si la queremos imprimir en una columna concreta podemos hacer PRINT TAB (10) A. Nos podemos complicar más la vida si lo que queremos es realizar un listado con una columna de cantidades numéricas alineadas por la derecha. En este caso la instrucción sería: PRINT TAB(10) RIGHT$(Z2$+STR$(A),15), siendo Z2$ una cadena con un mínimo de 15 espacios en blanco. La instrucción PRINT puede complicarse todo lo que queramos para efectuar una formatación concreta; pero llega un momento, por ejemplo cuando queremos poner puntos de separación entre los miles en que una sola instrucción PRINT no basta: hemos de fabricarnos una rutina.

¿Qué debe hacer la rutina? Le debemos entregar en una variable la cantidad numérica a formatar, la rutina debe devolver en una variable de cadena el resultado ya formatado y a una longitud determinada. Esta longitud la podemos facilitar en otra variable, llamada variable de control, pera que la rutina inserte los espacios en blanco necesarios por la izquierda para llegar a esta longitud. En una tercera variable podemos entregar el número de decimales que deseamos. Por ejemplo queremos formatar una cifra con puntos de miles, dos decimales y a una longitud máxima de 15 caracteres (incluyendo los puntos de miles y la coma decimal):

MX=15: MD=2: B=123456.7 : GOSUB 26200: PRINT "*"; B$; "*" debería provocar el siguiente resultado:

* 123.456'70* pero

las variables MX y MD pueden entregarse en una sola en que la parte entera indique la longitud de la cadena y la parte decimal el número de decimales. De esta manera el ejemplo anterior quedaría así:

MX=15.2 : B=123456.7 : GOSUB 26200 : PRINT "*"; B$; "*".

Obsérvese la línea 26299 de la rutina. Si efectuamos:

MX=15.2 : B=123456.7 : GOSUB 26299.

se formatea la variable y se imprime el resultado automáticamente, terminando con un punto y coma por si deben imprimirse más datos en esta misma línea. La impresión se realiza en el fichero número 4, que debe de estar abierto previamente como pantalla (OPEN 4,3) o impresora (OPEN 4,4).

Ejemplo

Vamos a ver un ejemplo completo. Haremos un listado con tres columnas: en la primera editaremos los números del 1 al 10, en la segunda el mismo número elevado a la séptima potencia, y en la tercera su logaritmo con cinco decimales:

```
10 Z2$=" ":FORX=1TO80: Z1$=Z2$+
   " ": NEXT
20 OPEN 4,4: REM LISTADO POR
   IMPRESORA
100 FOR N=1 TO 10
110 B=N: MX=5: GOSUB 26299
120 B=N7: MX=12: GOSUB 26299
130 B=LOG(N) : MX=10.5: GOSUB
    26299
140 PRINT #4: NEXT
150 END
```

El resultado sería el siguiente:

1	1	0'00000
2	128	0'69315
3	2.187	1'09861
4	16.384	1'38629
5	78.125	1'60944
6	279.936	1'79176
7	823.543	1'94591
8	2.097.152	2'07944
9	4.782.969	2'19722
10	10.000.000	2'30259

Explicación de la rutina

El funcionamiento exacto de la rutina, línea a línea, es el siguiente:

26205: Traspasa a MY la longitud de la cadena de salida y a MD el número de decimales deseados. A lo largo de toda la rutina MX no es afectado. Si no se indican decimales (IF MD=0...) salta a 26270 donde se hace un proceso especial más rápido para cuando no hay decimales.

26210: Inicializa X, que se usa en la línea 26235. Redondea B al número de decimales deseados y pone el signo en X$ (nulo o signo menos).



26215: Asigna a B1 la parte decimal de B. Asigna a B$ el valor entero...

26220: ... para eliminar el signo, que ya ha quedado en X$.

26225: Comprueba que el dato formateado no exceda de la longitud indicada como máxima.

26230: Si hay overflow rellena B$ con caracteres dolar y finaliza la rutina.

26235: Pone los puntos de miles cada tres dígitos.

26240: Si la parte decimal es cero, pone los ceros pertinentes en B1$

26245: Si hay decimales los asigna a B1$, fuerza los ceros correspondientes en la izquierda...

26255:... y complementa los necesarios por la derecha.

26260: Reúne en B$ el signo (X$), la parte entera con los puntos de miles ya puestos (B$) y la parte decimal también con la coma ya puesta (B1$). Además rellena por la izquierda los espacios en blanco necesarios (Z2$) hasta llegar a la longitud solicitada en MX. Fin de la rutina cuando se tratan decimales.

26270 a 26295: Formatación sin decimales. Ejecuta un proceso muy similar al anterior en lo que a la parte entera se refiere. Al no tener que tratar los decimales se hace mucho más rápido.

26299: Segunda posible entrada a la rutina. Formata el dato y lo envía por el fichero 4, que debe estar ya abierto para pantalla (OPEN 4,3) o impresora (OPEN 4,4).

Obsérvese que a lo largo de la rutina se utilizan una serie de variables de trabajo, además de las variables de entrada/salida B, MX, y B$. Si el programador utiliza alguna de estas variables en su programa, debe ser consciente que una vez haya llamado esta rutina de formatación los contenidos de las mismas quedaran alterados. Las variables utilizadas, junto con la función que realizan dentro de la rutina, son:

MY - Longitud de la cadena de salida.

MD - Número de decimales deseados.

X$ - Memoriza el signo.

B1 - Parte decimal del dato.

X - Contador de puntos de miles.

X1 - Auxiliar utilizada en la formatación de los decimales (26210 y 26215).

B2 - Auxiliar utilizada en las líneas 26215 y 26220.

El próximo mes... otra rutina!

Ejemplo

```
10 Z2$="":FORX=1TO80:Z2$=Z2$+"[SPC]"
:NEXT
15 ZI$="":FORX=1TO80:ZI$=ZI$+"0":NEX
T
20 OPEN4,4:REM LISTADO POR IMPRESORA

100 FORN=1TO10
110 B=N:MX=5:GOSUB26299
120 B=N↑7:MX=12:GOSUB26299
130 B=LOG(N):MX=10.5:GOSUB26299
140 PRINT#4:NEXT
150 END
```

Formatación numérica

```
26200 REM  * FORMAT.MILES **********
***
26205 MY=INT(MX):MD=INT(MX*10-MY*10)
:IFMD=0THEN26270
26210 X=3:X1=INT(10↑MD+.5):B=SGN(B)*
INT(ABS(B)*X1+.5)/X1:X$="":IFB<0THEN
X$="-"
26215 B2=ABS(B):B1=INT((B2-INT(B2))*
X1+.5):B$=STR$(INT(B2))
26220 B$=RIGHT$(B$,LEN(B$)-1)
26225 IFLEN(X$)+LEN(B$)+INT((LEN(B$)
-1)/3)+MD+1<=MYTHEN26235
26230 B$="":FORX=1TOMY:B$=B$+"$":NEX
T:RETURN
26235 IFLEN(B$)>XTHENB$=LEFT$(B$,LEN
(B$)-X)+"."+RIGHT$(B$,X):X=X+4:GOTO2
6235
26240 IFB1=0THENB1$="'"+LEFT$(ZI$,MD
):GOTO26260
26245 B1$=STR$(B1):B1$="'"+RIGHT$("0
00000000"+RIGHT$(B1$,LEN(B1$)-1),MD)

26255 B1$=LEFT$(B1$+"000000000",MD+1
)
26260 B$=RIGHT$(Z2$+X$+B$,MY-MD-1)+B
1$:RETURN
26270 B=SGN(B)*INT(ABS(B)+.5):B$=STR
$(ABS(B)):B$=RIGHT$(B$,LEN(B$)-1)
26275 IFLEN(B$)>3THENB$=LEFT$(B$,LEN
(B$)-3)+"."+RIGHT$(B$,3)
26280 IFLEN(B$)>7THENB$=LEFT$(B$,LEN
(B$)-7)+"."+RIGHT$(B$,7)
26285 IFB<0THENB$="-"+B$
26290 IFLEN(B$)>MYTHENB$="":FORX=1TO
MY:B$=B$+"$":NEXT:RETURN
26295 B$=RIGHT$(Z2$+B$,MY):RETURN
26299 GOSUB26200:PRINT#4,B$;:RETURN
```

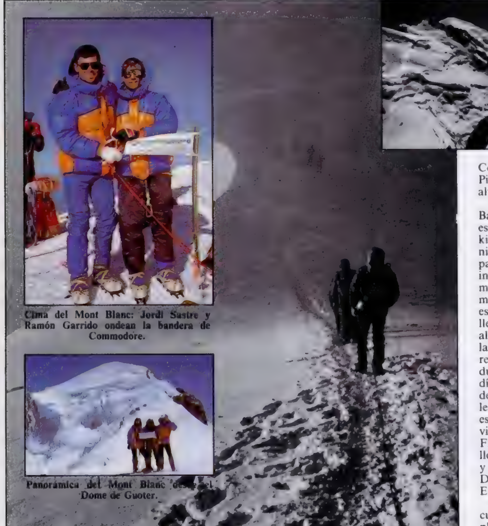
Commodore en la cima de Europa



Instantánea cerca de la cumbre tomada durante la ascensión.

Aprovechando las vacaciones y para huir temporalmente del calor veraniego, un equipo de MICROELECTRONICA Y CONTROL, S.A. se dispuso a respirar aires más frescos. Su objetivo era nada más y nada menos que escalar y hacer ondear la bandera de COMMODORE COMPUTER en la cima del Mont Blanc, el pico más alto de Europa. ¡Y lo consiguieron! Nos han entregado las fotos que reproducimos como testimonio del gran éxito. Sirvan también para satisfacción de todos nosotros viendo nuestra marca favorita en la cúspide de Europa.

Como sabéis, el macizo del Mont Blanc forma parte de la gran cordillera de los Alpes. Esta situación en la frontera entre Francia e Italia, muy cerca de Suiza. Al pie del mismo se halla el pueblo francés de Chamonix, famoso por ser el punto de partida de todas las expediciones que acceden al Mont Blanc por su vertiente francesa. Los que lo hacen por la italiana salen de Courmayeur. El Mont Blanc es un paraíso de roca, hielo y toma el nombre de su pico más alto, el Mont Blanc, que con sus 4.8Ø7 metros de altura sobre el nivel del mar se erige además como el más alto de todos los Alpes y, por tanto, también de todo el continente europeo.



Cima del Mont Blanc: Jordi Sastre y Ramón Garrido ondean la bandera de Commodore.

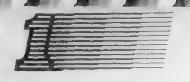
Panorámica del Mont Blanc desde el Dome de Guoter.

Panorámica del Mont Blanc desde la Aiguille du Midi.

Como referencia, el pico más alto de los Pirineos, el Aneto, tiene 3.4Ø4 metros de altura.

Nuestros expedicionarios partieron de Barcelona el día 28 de julio, recorriendo este mismo dia (en coche, no a pie) los 84Ø kilómetros que los separaban de Chamonix. Tras un día entero de descanso y preparativos y, cómo no, de algo de turismo, inician la ascensión el lunes, día 30, por la mañana. Los primeros metros los salvan mediante una telecabina, por lo que la escalada como tal no empieza hasta que llegan a Nid d'Aigle, a 2.3ØØ metros de altura. De aquí ascienden cargados con las mochilas y todo el material hasta el refugio de Tete Rousse (3.167 m.), donde duermen la primera noche. El segundo día lo invierten en llegar hasta el refugio de Gouter (3.817 m). A causa de una violenta tormenta no pueden culminar la escalada el tercer día, como estaba previsto, sino que deben esperar un día más. Finalmente, el día 2 de agosto consiguen llegar a la cumbre del Mont Blanc (4.8Ø7) y hacer ondear la bandera de COMMODORE COMPUTER en lo más alto de Europa.

Reciban nuestas felicitaciones, y a ver si cunde el ejemplo y alguien se anima a plantar nuestra bandera en... el Everest (¡ejem!), por citar un ejemplo...



FIRST S.A.

C/ Aribau, 62. BARCELONA- 08011
Tel. (93) 323 03 90
Tlx 53947 FIRS E (ESPAÑA)

Ordenadores más Personales!



SEGUIMOS BUSCANDO DISTRIBUIDORES



SE COMPLACE EN ANUNCIAR LA DISTRIBUCION DE PRODUCTOS

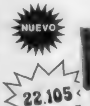
commodore

DE: microelectrónica y control, s.a. MEC

IMPORTADOR EXCLUSIVO PARA ESPAÑA



ErgoVisión
(Base móvil)
29.800 ptas.



KoalaPad Touch Tablet (disco)



22.105 PTAS

para **COMMODORE 64**

Auténtico tablero gráfico de dibujo para su ordenador. Conectado puede hacer sus dibujos como lo hace con su lápiz y papel. Incluye el programa MICRO ILLUSTRATOR de tecnología LISA, permite Dibujar, Puntos, Líneas, Rayas, Rellenar, Cuadrados, Círculos, Discos, Rellenar, Borrar, Magnificar. Guardará sus gráficos y mucho más. Pásese y compruébelo Vd. mismo



FIRST S.A. importador para españa de Koala

METAMORPHIC
25.000 ptas.



COMMODORE 64
+ 2 SUPER JOYSTICK
+ .... = PTAS.

FIRST VENDEMOS IMPRESORAS MAS PERSONALES

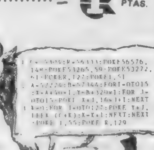
star

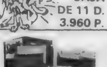


MICROANGELO

GEMINI 10 X 79.500 Ptas
GEMINI 15 X 115.000 Ptas
INCLUYE LA INTERFACE MICROANGELO Y CONECTORES (OFERTA SOLO PARA ESTE MES)

LLAME PARA SOLICITAR SU PRECIO SI NO PRECISA INTERFACE
LLAME PARA SOLICITAR SU PRECIO PARA EL RESTO DE IMPRESORAS DE LA GAMA STAR. TAMBIEN INCLUYEN LA TARJETA MICROANGELO



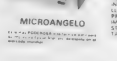



LAS TARJETAS DE FIRST

- TARJETA VIC 20 -40/80 COL. LLAME ☎
- TARJETA VIC 20 -MEMO. 64 K. LLAME ☎
- SLOTS EXPANSION V20 Y C64 LLAME ☎
- TARJETA COMMO 64 80 COL. LLAME ☎

Más de 200 programas para su VIC 20 y COMMODORE 64:
- Utilidades - Lenguajes
- Sis. Operativos - Aplicaciones
- Juegos (nacionales e importación)
(Amplio surtido de programas educativos para sus hijos)

- CASSETE COMMODORE LLAME ☎
- INTERFACE RS 232 LLAME ☎
- PADDLES LLAME ☎
- FUNDA COMMODORE 64 LLAME ☎




Datalife
10 DISKETTES + 2 DISKETTES (LIMPIA CABEZAL)
5.300 ptas

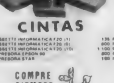
SIEMPRE NUEVAS IDEAS EN FIRST NEWS Y LO RECIBO GRATUITAMENTE INCREIBLE

FABULOSO REGALO TIP POSTER

A todos los propietarios de un VIC 20 ó C 64, manden un sobre indicando: Tipo de Ordenador, anualidad y lugar donde lo compró, tipo de configuración de su equipo. Recibirán: un FABULOSO TIP POSTER COMMODORE de FIRST. Manden su carta a FIRST TIP POSTER. Y por favor, dennos tiempo a contestar todas sus cartas.



SUPER JOYSTICK
COMPATIBLE CON COMMODORE 64, VIC 20, ATARI, SPECTRAVIDEO
2.200!!



CINTAS

CINTA CASSETTE INFORMATICA F20 (1)	135 Pts
CINTA CASSETTE INFORMATICA F20 (6)	600 Pts
CINTA CASSETTE INFORMATICA F20 (10)	1.100 Pts
CINTA IMPRESORA EPSON 80	800 Pts
CINTA IMPRESORA STAR	180 Pts



COMPRE SIEMPRE EN FIRST

PAPEL ESPECIAL PARA VD.

2.500 Hojas de papel (bordes perforados) Superblanco Grueso 80g	4.456 Ptas
2.500 Recibos negociables standard	4.683 Ptas
1.000 Etiquetas autoadhesivas (2 + líneas)	829 Ptas

Todos los envíos de papel son a portes debidos

Magic Desk

Ud y MAGIC DESK, penetrarán en la nueva generación de software. Imagínese usando su ordenador para hacer archivos y editar letras personales, etc. sin aprender un simple comando. Los comandos son GRAFICOS. Mueva sólo la mano animada y realice su deseo, listo para funcionar.
MAGIC DESK en FIRST............ LLAME ☎

First In Quality Software

LOS LIBROS DE FIRST

COMMODORE-64

Curso de Introducción al BASIC parte I	3.000
Curso de Introducción al BASIC parte II	3.000
Guía de Referencia del Programador	3.000
COMMODORE 64 COMPUTING	1.900

VIC-20

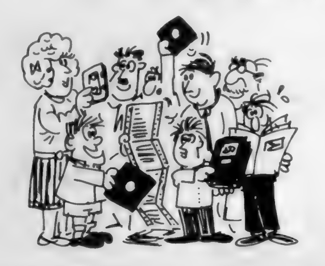
VIC 20 GUIA DEL USUARIO	2.300
LEARN TO USE THE VIC-20	1.600
VIC-20 EXPOSED	1.900



VENTA POR CORREO

ENTREGA INMEDIATA A PROVINCIAS

COMPARTIENDO EXPERIENCIAS ENTRE AMIGOS



Esta sección está dedicada a la colaboración de todos nuestros lectores y está dividida en dos partes:

1) Programación: Programas y similares.

2) Magia: Trucos, sugerencias, etcétera *(pág. 54)**.*

Habrá premios y alicientes para todos los participantes *(ver pág. 6)**.*

Enviarnos vuestra dirección para que podáis poneros en contacto unos con otros. ¡Animaros, chicos (...y chicas)!

Editor

C-64

Pablo Cambra Brown
Martínez Valls, 14. P3
Onteniente (Valencia)

Como indica el nombre de este programa, consiste en una ayuda al programador tanto en cuanto en que sirve para calcular los números que van ligados a un carácter programado por software.

Las teclas que comandan las distintas órdenes que posee el programa son:

D—(dibujar) rellenan un hueco del cuadriculado 8×8 que aparece en pantalla.

B—(borrar) vacía un hueco del cuadriculado que había anteriormente llenado.

C—(compilar) calcula y exhibe las cantidades que se trascienden de la figura dibujada como una línea de DATA con número de línea anteriormente definida por el usuario. Cuando el usuario usa esta función deberá bajar el cursor (con las TECLAS DE CURSOR) hasta colocarlo sobre la línea de DATA y apretar RETURN dos veces. Posteriormente podrá definir un nuevo carácter si lo desea, siempre y cuando elija un número de línea de DATA que NO SEA 1 NI mayor que 900 NI igual al número de línea de DATA anterior.

Si el usuario no elige crear un nuevo carácter, **TODO** el programa se borrará menos la instrucción 1 y las líneas de data que definen los caracteres.

Borre la línea 1 apretando HOME y RETURN.

Las líneas de DATA están ya listas para mezclar ("merge") con otro programa.

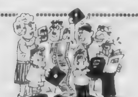
El asterisco señala la posición actual sobre el cuadriculado cuando se está dibujando el gráfico y las teclas del cursor lo mueven por el cuadriculado sin que afecte en absoluto lo dibujado.

```
1 POKE650,128:GOTO999
990 DE=PEEK(61)+256*PEEK(62)+3:POKE2
52,INT(DE/256):POKE251,DE-256*PEEK(2
52)
991 POKEDE-2,0:POKEDE-1,0:POKE45,PEE
K(251):POKE46,PEEK(252):END
999 DIMJ(8,8):FORI=0TO7:FORH=0TO7:J(
I,H)=79:NEXTH,I
1000 INPUT"[CLR]STARTING[SPC]DATA[SPC]
LINE[SPC]NUMBER";LIN
1010 IFLIN>990GOTO1000
1020 PRINT"[CLR]"
1030 FORI=0TO7:PRINT"[SPC][8SHIFTO][COMMH]
":NEXT
1040 PRINT"[SPC][8COMMY]"
1050 A=1065:CL=0:RO=0
1060 POKEA,170:B=A
1070 GETA$:IFA$=""THEN1070
1080 IFA$="[CRSRU]"THENA=A-40:POKEB,
J(CL,RO):RO=RO-1:IFA<1065THENGOSUB11
80:GOTO1060
1090 IFA$="[CRSRD]"THENA=A+40:POKEB,
J(CL,RO):RO=RO+1:IFA>1352THENGOSUB11
90:GOTO1060
```

```
1100 IFA$="[CRSRL]"THENA=A-1:POKEB,J
(CL,RO):CL=CL-1:IFCL<0THENA=A+8:CL=7
:GOTO1060
1110 IFA$="[CRSRR]"THENA=A+1:POKEB,J
(CL,RO):CL=CL+1:IFCL>7THENA=A-8:CL=0
:GOTO1060
1120 IFA$="D"THENPOKEA,160:J(CL,RO)=
160:GOTO1160
1130 IFA$="B"THENPOKEA,79:J(CL,RO)=7
9:GOTO1160
1140 IFA$="C"THENPRINT"[CLR]":GOTO30
00
1150 GOTO1060
1160 CL=CL+1:A=A+1:IFCL>7THENA=A-8:C
L=0
1170 GOTO1060
1180 A=A+320:RO=7:RETURN
1190 A=A-320:RO=0:RETURN
3000 LIN$=STR$(LIN):PRINT"[4CRSRD][CRSRL]
"LIN$;"DATA";
3005 FORH=0TO7:DT=0
3010 FORI=7TO0STEP-1
3020 IFJ((7-I),H)=160THENDT=DT+(2↑I)
```

```
3030 NEXT
3031 IFDT<10THENG=1:GOTO3034
3032 IFDT<100THENG=2:GOTO3034
3033 G=3
3034 DT$=RIGHT$(STR$(DT),G):PRINTDT$
,",";:NEXT
3038 PRINT"[CRSRL][SPC]":PRINT"RUN30
50"
```

```
3040 PRINT"[HOM]":END
3050 INPUT"[CLR]QUIERE[SPC]DEFINIR[SPC]
MAS[SPC]CARCTERES";R$
3060 IFR$<>"NO"THENIFR$<>"SI"THEN305
0
3070 IFR$="SI"THENRUN999
3080 PRINT"[CLR]1":GOTO990
```



Verbos

C-64

Manuel López i Serres
C/ Sant Roc, 7 - 2º 1.ª
Maspujols
(Tarragona)

Hola amigos de Commodore:

Antes de hablaros del programa que os envío deseo deciros que estoy muy contento con vuestra revista, pero que yo os agradecería que entre los programas que editáis hubieran más dedicados a los estudiantes, yo os animo a vosotros y al resto de asiduos, entre los cuales creo que hay bastantes estudiantes, que escriban programas de matemáticas, gráficas, física, química, etc...

Hace muy poco que tengo mi Commodore 64 y de momento sólo me funcionan correctamente unos pocos programas, del que he escogido este porque es lo que a mí me gusta menos estudiar. El programa sirve para aprender verbos, en el listado que os envío hay el verbo Sum, que he de saberme para aprobar Latín de segundo, aunque ya he terminado C.O.U.

El programa se basa en ir entrando el tiempo verbal que te pide y te va diciendo a medida que escribes formas si están bien o mal, al final te da la nota. Para aprender otro verbo es necesario cambiar las sentencias data sin olvidar poner una C al final de cada una y una D al final de la última. Seguramente el programa puede ser mejorado, yo he pensado en ponerle música, pero mis conocimientos no me llegan, si alguien lo mejora ruego que me lo comunique, mis señas son: Manuel López i Serres. C/ Sant Roc, 7 Maspujols-Tarragona. Telf.: 81-51-94.

Ahora os explico un poco las líneas del programa:

En la 10 dimensiona la matriz B$ en 12, para almacenar los tiempos verbales.

En la 20 igualo la J y X a 0 para hacerlas servir como control de fallos y control de tiempos verbales. En la 70 se almacenan los modos verbales. En la 70 controlamos y visionamos los modos verbales.

Las líneas 90 a 93 clasifican los tiempos verbales y la 100 los controla e imprime en pantalla.

La 120 lee la forma verbal correcta y comprueba si es el final de una línea.

En la 122 comprueba si hay más formas.

En la 130 se nos pide una palabra, si coincide se ve "be" (bien) y si no "malament" (mal) junto con el total de errores (línea 155).

En las líneas 160 a 198 se encuentran las formas verbales correctas en la 200 se calcula la nota.

Para que se ejecute otro verbo es necesario cambiar las sentencias DATA en el orden correcto y poner una C al final de cada una o una D si es la última, además en otros idiomas cambian el nombre de los tiempos verbales o hay más como por ejemplo en el catalán, en tal caso hay que hacer los ajustes oportunos.

```
10 DIMB$(12)
20 J=0:X=0
70 C$(0)="INDICATIU":C$(1)="SUBJUNTI
U":C$(2)="IMPERATIU"
80 PRINT"[CLR]":FORY=0TO2:PRINT"[6CRSRR]
[RVSON]";C$(Y),"[RVSOFF]"
90 B$(0)="PRESENT":B$(1)="PRETERIT[SPC]
IMPERFECTE":B$(2)="FUTUR[SPC]SIMPLE"

91 B$(3)="PRETERIT[SPC]PERFECTE":B$(
4)="PRETERIT[SPC]PLUSCUAMPERFECTE"
92 B$(5)="FUTUR[SPC]ANTERIOR":B$(6)=
B$(0):B$(7)=B$(1):B$(8)=B$(3):B$(9)=
B$(4)
93 B$(10)=B$(0):B$(11)=B$(2)
100 PRINT"[CRSRR][RVSON]";B$(X);"[RVSOFF]
":X=X+1
120 READA$:IFA$="C"THENGOTO150
122 IFA$="D"THENGOTO200
130 INPUTA1$:IFA1$<>A$THENGOTO155
140 PRINT"[CRSRU][24CRSRR][RVSON]BE[RVSOFF]
":GOTO120
150 IFX=6ORX=10THENNEXTY
151 IFX<12THENGOTO100:END
155 J=J+1:PRINT"[CRSRU][24CRSRR][RVSON]
MALAMENT[RVSOFF]";J:GOTO130
```

```
160 DATASUM,ES,EST,SUMUS,ESTIS,SUNT,
C
170 DATAERAM,ERAS,ERAT,ERAMUS,ERATIS
,ERANT,C
180 DATAERO,ERIS,ERIT,ERIMUS,ERITIS,
ERUNT,C
190 DATAFUI,FUISTI,FUIT,FUIMUS,FUIST
IS,FUERUNT-FUERE,C
191 DATAFUERAM,FUERAS,FUERAT,FUERAMU
S,FUERATIS,FUERANT,C
192 DATAFUERO,FUERIS,FUERIT,FUERIMUS
,FUERITIS,FUERINT,C
193 DATASIM,SIS,SIT,SIMUS,SITIS,SINT
,C
194 DATAESSEM,ESSES,ESSET,ESSEMUS,ES
SETIS,ESSENT,C
195 DATAFUERIM,FUERIS,FUERIT,FUERIMU
S,FUERITIS,FUERINT,C
196 DATAFUISSEM,FUISSES,FUISSET,FUIS
SEMUS,FUISSETIS,FUISSENT,C
197 DATAES,ESTE,C
198 DATAESTO,ESTO,ESTOTE,SUNTO,D
200 N=((66-J)/66)*10:PRINT"[RVSON]NO
TA:[RVSOFF][SPC]";N:END
```



Quinielas

C-64

Alejandro Cantó
C/ Doctor Gadea, 1-2º E+F
Alicante 03003

Explicación del programa.

Líneas 5, 10 y 25. Define variables.

20-80 Dibuja boleto.

90-110 Pregunta probabilidades para las tres variables.

120-160 Calcula resultados y los imprime en el boleto.

200-230 Comprueba si queremos nuevo boleto.

250-final Presentación.

Las variables más importantes son:

—J. Comprueba si queremos parar el programa.

—A$, B$, C$. Dibujan el boleto.

—A1, A2, AX. Probabilidades.

—6$. Dibuja la presentación.

—PO$. Imprime los resultados en el boleto.

El programa está hecho de forma que si al presionar RETURN, después de cada INPUT las probabilidades son para:

'1' — 50%
'2' — 25%
'X' — 25%

```
5 J=0:GOSUB250
10 A$="[COMM7][COMMA][SHIFTC][COMMR]
[SHIFTC][COMMR][SHIFTC][COMMS]":B$="
[SHIFTB][SPC][SHIFTB][SPC][SHIFTB][SPC]
[SHIFTB]":C$="[COMMZ][SHIFTC][COMME]
[SHIFTC][COMME][SHIFTC][COMMX]"
20 PRINT"[CLR]",TAB(14)"[RED][RVSON]
QUINIELAS":PRINTTAB(15)"[COMM4][CRSRD]
[SPC]1[SPC]X[SPC]2[SPC]"
25 PO$="":A1=51:A2=30:AX=19
30 PRINTTAB(15)A$:FORP=1TO14:PRINTTA
B(15)B$:NEXT:PRINTTAB(15)C$
40 PRINT"[HOM][3CRSRD]":FORP=1TO14
50 IFP<10THEN70
60 GOTO80
70 PRINT"[COMM6]";P;"[COMM6]--------
---":NEXT
80 PRINTP;"[COMM6]----------":NEXT
90 INPUT"[COMM8][CRSRD]PROBABILIDAD[SPC]
DE[SPC]'1'[SPC]EN[SPC]%";A1
100 INPUT"[CRSRD]PROBABILIDAD[SPC]DE
[SPC]'X'[SPC]EN[SPC]%";AX
110 INPUT"[CRSRD]PROBABILIDAD[SPC]DE
[SPC]'2'[SPC]EN[SPC]%";A2
120 IFA1+A2+AX<>100THEN20
130 FORP=1TO14:PR=INT(RND(0)*100)+1:
PO$=PO$+"[CRSRD]":A2=(A1+AX)
140 IFPR<=A1THENPRINT"[HOM][3CRSRD]"
,PO$,"[4CRSRL][CYN]1":NEXT:GOTO200
150 IFPR>=A2THENPRINT"[HOM][3CRSRD]"
,PO$,"[CYN]2":NEXT:GOTO200
160 PRINT"[HOM][3CRSRD]",PO$,"[2CRSRL]
[CYN]X":NEXT:GOTO200
200 PRINT"[HOM][4CRSRD]",TAB(20),"[WHT]
OTRA?"
210 GETA$:IFA$="S"THEN10
220 IFA$<>"N"THEN210
230 J=9:GOSUB250
250 POKE53280,0:POKE53281,0:PRINT"[YEL]
[CLR]PROGRAMADO[SPC]POR[SPC]ALEJANDR
O[SPC]CANTO[SPC]&[SPC]COMPANY"
260 PRINT:PRINT:PRINTTAB(12)"PARA[SPC]
COMMODORE[SPC]WORLD[10CRSRD]"
290 IFJ=9THENEND
300 G$="[RVSON][SHIFT£][SPC][COMM*][RVSOFF]
[SPC][RVSON][SPC][RVSOFF][SPC][RVSON]
[SPC][RVSOFF][SPC][RVSON][SPC][RVSOFF]
[SPC][RVSON][SPC][RVSOFF][SPC][RVSON]
[SPC][RVSOFF][SPC][RVSON][SPC][RVSOFF]
[SPC][RVSON][3SPC][RVSOFF][SPC][RVSON]
[SPC][RVSOFF][3SPC][RVSON][3SPC][RVSOFF]
[SPC][RVSON][SHIFT£][SPC][RVSOFF][SHIFT£]
":GOSUB330
310 G$="[RVSON][SPC][RVSOFF][SPC][RVSON]
[SPC][RVSOFF][SPC][RVSON][SPC][RVSOFF]
[SPC][RVSON][SPC][RVSOFF][SPC][RVSON]
[SPC][RVSOFF][SPC][RVSON][SPC][RVSOFF]
[COMM8][RVSON][SPC][RVSOFF][SPC][RVSON]
[SPC][RVSOFF][SPC][RVSON][SPC][COMMT]
[RVSOFF][2SPC][RVSON][SPC][RVSOFF][3SPC]
[RVSON][SPC][COMMU][SPC][RVSOFF][SPC]
[COMM*][RVSON][SPC][COMM*]":GOSUB330
320 G$="[COMM*][RVSON][SPC][RVSOFF][SHIFTZ]
[SPC][RVSON][3SPC][RVSOFF][SPC][RVSON]
[SPC][RVSOFF][SPC][RVSON][SPC][RVSOFF]
[SPC][RVSON][SPC][RVSOFF][SPC][RVSON]
```

GLOSARIO

ACU.—(Automatic calling unit), la unidad de llamada automática es un dispositivo que permite que nuestro modem marque automáticamente el número de teléfono de la red con la que queremos conectar. De este modo nos evitamos tener que marcar a mano una y otra vez esperando que deje de estar comunicando.

A/D.—Son las siglas de Analógico/Digital, indican la conversión de uno a otro sistema. En nuestros equipos unos conversores A/D se encargan de leer la posición de los mandos de juego tipo paddle.

ADA.—Es un lenguaje de alto nivel para procesos en tiempo real. Fue desarrollado en 1983 por un científico de la universidad de Nueva York para el departamento de defensa de los Estados Unidos. Su nombre es el de Augusta Ada Byron, hija de Lord Byron, que se considera la primera programadora del mundo.

ADC.—Conversor analógico/digital (ver A/D).

Add-on.—Accesorios, son circuitos o sistemas periféricos que al conectarlos al ordenador incrementan su memoria o prestaciones.

ADDR.—Puede ser la abreviatura de dirección, direccionamiento, registro de dirección o sumador.

AL.—(Assembler Languaje), lenguaje ensamblador. Es el lenguaje más cercano al código máquina que todavía puede ser comprendido por el ser humano, ya que utiliza códigos mnemónicos que recuerdan fácilmente la función del cada instrucción.

Alfanumérico.—Relativo a juegos de caracteres que contienen letras, números y normalmente otros caracteres como símbolos de puntuación.

ALGOL.—(ALGOrithmic Language), es un lenguaje utilizado principalmente para expresar programas de ordenador con algoritmos.

```
[SPC][RVSOFF][SPC][RVSON][SPC][2COMMT]
[RVSOFF][SPC][RVSON][3SPC][RVSOFF][SPC]
[RVSON][SPC][RVSOFF][SPC][RVSON][SPC]
[RVSOFF][SPC][RVSON][SHIFT£][SPC][RVSOFF]
[SHIFT£]":GOSUB330
325 GOTO355
330 G$="[4SPC]"+G$
340 FORY=1TOLEN(G$)
350 PRINTRIGHT$(G$,Y);"[CRSRU]":NEXT
```

```
:PRINT:RETURN
355 PRINTTAB(12)"PULSA[SPC]UNA[SPC]T
ECLA"
360 FORLA=55896TO56015:GETA$:IFA$=""
THEN370
365 RETURN
370 POKELA,Z:NEXT:Z=Z+1:IFZ=255THENR
ETURN
380 GOTO360
```



Super Basic

VIC-20
sin o con ampliación

Marcos Gregori
Camping "CACTUS".
Peñiscola (Castellón)

Os envío en programa escrito en código máquina que permite introducir algunas variantes al BASIC del VIC-20.

La primera parte del programa, y la más importante, es una reproducción de una corta rutina situada en la ROM BASIC. Esta se encarga de gestionar los saltos a las distintas sentencias del BASIC. La primera parte se compone de dos bloques, la rutina y una tabla de saltos. La tabla contiene la dirección de llamada de las diferentes sentencias.

La sustitución de la rutina original por esta se consigue sin más que variar un puntero que está en las direcciones 0308h-0309h de forma que apunte al programa.

Como la nueva tabla está en RAM podemos efectuar en ella los cambios que queramos, para que apunten a posiciones de RAM en donde podemos colocar variantes de la instrucción. Este primer bloque ocupa 200 bytes (direcciones 1000 h a 10C9).

A modo de ejemplo he incluido 4 nuevas sentencias que no son más que variantes de otras ya existentes:

PRINT@x, y "comentarios". Imprime en cualquier parte de la pantalla (indicado por x e y) los clásicos PRINTS. Se puede utilizar, por lo demás, con todas las variantes del PRINT original.

INPUT@x, y "comentarios"; lista de variables. Lo mismo aplicado a la sentencia INPUT.

LOAD#"nombre del programa". Mezcla dos o más programas cargados desde el DATASSETTE. La única condición es que los números de línea sean distintos. En el caso de que se mezclen más de dos programas hay que tener cuidado con el orden de la numeración, porque el VIC no lo reestablece.

RESTORE (número de línea). Es un RESTORE calculado a una determinada línea data del programa. Admite variables y expresiones como número de línea. Utilizado con tablas DATA grandes permite acceder a cualquier dato mucho más rápidamente, pues no se tiene que leer cada vez la tabla desde el principio.

Y eso es todo.

Si os decidís a fabricar vuestras propias variaciones tened en cuenta lo siguiente: Para que el salto se efectúe tiene que haber una sentencia original del BASIC del Vic, es decir, que si queréis introducir, por ejemplo, un comando que cambie los colores del borde, pantalla y caracteres, debéis precederlo de una sentencia existente. Una buena solución sería DEF COLOR a, b, c. La sentencia DEF es reconocida independientemente de FN, y por lo tanto se puede interceptar.

En el punto de la nueva entrada una manera de examinar si existe o no variación sobre la sentencia original es:

```
Entrada  JSR 0079
         CMP Nuevo carácter
         BEQ Variación
         JMP Comando normal
Variación....
.........
```

Este método permite sólo detectar un carácter, para detectar expresiones enteras, se debe programar un sistema más elaborado.

A continuación incluyo una lista de las sentencias en el mismo orden en que aparecen en la tabla para que podáis interpretar cualquier comando:

END,FOR,NEXT,DATA,INPUT:INPUT,DIM,READ,LET,GOTO,RUN,IF,RESTORE,GOSUB,RETURN,REM,STOP,ON,WAIT,LOAD,SAVE,VERIFY,DEF,POKE,PRINT,PRINT,CONT,LIST,CLR,CMD,SYS,OPEN,CLOSE,GET,NEW,SGN,INT,ABS,USR,FRE,POS,SQR,RND,LOG,EXP,COS,SIN,TAN,ATN,PEEK,LEN,STRS,VAL,ASC,CHRS,LEFTS,RIGHTS,MIDS,GO.

Si decidís publicar mi carta sería interesante que incluyerais un listado en ensamblador del mismo programa y un DUMP de la tabla de saltos, yo no os lo puedo enviar porque no tengo impresora.

Bueno, eso es todo. Me despido esperando que encontréis el programa interesante y que programéis un montón de nuevos comandos para vuestro VIC.

P.D.: Me olvidaba de deciros que el programa está situado en las direcciones 1000 a 11A5 y que por lo tanto ***los que dispongan de ampliación de más de 8K*** tienen que introducir antes POKE 641,0:POKE 642,32:POKE 648,30:SYS 64824. Esto aísla 3.5 K contra sobreescritura por el BASIC. ***Los que utilicen un VIC estándar*** pueden probar con POKE 641,0:POKE 642,18:SYS 64824, esto les quitará unos 500 bytes al principio de la memoria del usuario. (Yo no tengo ese tipo de problemas ¡¡¡Mi VIC tiene 8K de RAM, claro, entre las direcciones A000 y BFFF!!!

```
0 REM
1 REM******************
2 REM* SUPER BASIC    *
3 REM*MARCOS GREGORI*
4 REM*    14:7:84     *
5 REM******************
6 REM
7 REM
8 FORT=4096TO4515:READA:POKET,A:NEXT
```

```
9 SYS4096
10 DATA162,14,142,8,3,162,16,142,9,3
,96
11 DATA108,8,3,32,115,0,32,23,16,76,
174
12 DATA199,240,60,233,128,144,17,201
,35,176,23
13 DATA10,168,185,87,16,72,185,86,16
,72,76
```

```
14 DATA115,0,76,165,201,201,58,240,2
14,76,8
15 DATA207,201,75,208,249,32,115,0,1
69,164,32
16 DATA255,206,76,160,200,56,165,43,
233,1,164
17 DATA44,176,1,136,133,65,132,66,96
,48,200
18 DATA65,199,29,205,247,200.164,203
,249.16,128
19 DATA208,5,204,164,201,159,200,112
,200.39,201
20 DATA12,17,130,200,209,200.58,201,
46,200,74
21 DATA201,44,216,123,17,82,225,97,2
25,178,211
22 DATA35,216,127,202,233,16,86,200,
155,198,93
23 DATA198,133,202,38,225,186,225,19
5,225,122,203
24 DATA65,198,57,220,204.220,88,220,
0,0,125
25 DATA211,158,211,113,223,148,224,2
34,217,237,223
26 DATA97,226,104,226,177,226,11,227
,13,216,124
27 DATA215,101,212,173,215,139,215,2
36,214,0,215
28 DATA44,215,55,215,32,155,215,134,
251,32,241
29 DATA215,134,252,224,22,176,6,165,
251,201,23
30 DATA144,3,76,72,210,168,24,32,240
,255,32
```

```
31 DATA253,206,96,32,121.0,201,64,24
0,3,76
32 DATA160,202,32,202,16,76,160,202,
32,121,0
33 DATA201,64,240,3,76,191,203,32,20
2,16,32
34 DATA121,0,76,191,203,32,121,0,201
.40.240
35 DATA3,76 29,200,32,241,206,32,247
,215,165
36 DATA20,133,63,165,21,133,64,56,32
,156,255
37 DATA134,251,132,252,160,3,177,251
,197,63,208
38 DATA17,160,4,177,251,197,64.208,9
,165,251
39 DATA133,65,165,252,133,66,96,160,
4,177,251
40 DATA197,64,144,13,208,39,160,3,17
7,251,197
41 DATA63,144,3,76,119,17,160,2,177,
251,133
42 DATA254,160,1,177,251,133,253,56,
165,253,233
43 DATA1,133,251,165,254,233,0,133,2
52,76,45
44 DATA17,162,17,76,55,196,32,121,0,
201,35
45 DATA240,3,76,101,225,230,122,165,
45,56,233
46 DATA2,133,251,165,46,133,252,169,
0,133,10
47 DATA32,209,225,165,10,166,251,164
,252,76,114
48 DATA225,114
```



Dibujo/Teclado

C-64

J.M. González Uriarte
C/ Umbe, 5-5º c-drcha.
Telf.: (94) 469 05 69
Guecho (Vizcaya)

Me dirijo a vosotros con la familiaridad que supone el ser suscriptor de la revista y estar disfrutando con los diferentes artículos que presentáis cada mes y que son enseñanzas prácticas que, poco a poco, dan un mayor conocimiento sobre las prestaciones de los ordenadores personales y que hacen vislumbrar las diferentes aplicaciones a las que cada uno, con su imaginación, pueda llegar.

Bajo mi punto de vista, el enfoque que dais a la revista, dirigida a principiantes y usuarios de ordenadores personales, no profesionales, es buena y os animo por ello a continuar este camino y mejorarlo, si cabe.

Dentro de la malla de puntos de alta resolución (320×200) se puede desplazar un punto inicial que se debe definir al principio por sus coordenadas (x: valor hasta 320/y: valor hasta 200) y que se desplaza pulsando las teclas I ↑, K ↓, J ←, L →, o bien, en diagonal pulsando alternativamente I-L, K-L, K-J, J-I.

Si se desea saltar a otro punto, debe pulsarse la tecla B. Aparecerán en pantalla las coordenadas del último punto ocupado por el cursor y que sirve de referencia, así como una serie de opciones a elegir; comenzar por otro punto, borrando lo anterior o permaneciendo en pantalla lo dibujado anteriormente. En ambos casos se pierden las nuevas coordenadas.

Comentarios del listado.

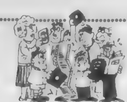
- 1- 4 Presentación.
- 5- 20 Petición de datos. (En 4 y 7 se establece un contador que se utilizará posteriormente para borrar, o no, el "BIT MAP").
- 35- 70 Protocolo de entrada en alta resolución. (En 55 se determina el condicional mediante el contador).
- 75 Dibuja el punto pedido.
- 85 Inserta la repetición de cada tecla pulsada.
- 90- 140 Dibuja cada punto según la tecla pulsada.
- 1000-1600 Bucle para dibujar en alta resolución.
- 2100 Para salir de alta resolución.
- 2200 Para entrar en la forma "carácter standart" de la pantalla.
- 2300-2530 Presentación.

```
1 REM***** DIBUJO DESDE EL TECLADO *
****
2 PRINT"[CLR]":PRINT:PRINT:PRINT:PRI
NT:PRINT
3 PRINTTAB(8);"[RVSON]DIBUJO[SPC]DES
DE[SPC]EL[SPC]TECLADO[RVSOFF]"
4 FOR W=0 TO 1000:NEXT W :C=0
5 PRINT"[CLR]":PRINT:PRINT:PRINT:PRI
NT:PRINT
```

```
7 C=C+1
10 INPUT"COORDENADAS[SPC]PUNTO[SPC]C
OMIENZO";X1,Y1
20 PRINT"[CLR]":PRINT:PRINT:PRINT:PR
INT:PRINT:PRINT
30 PRINTTAB(10);"ESPERE[SPC]UN[SPC]M
OMENTO"
35 FOR W=0 TO  500:NEXT W:PRINT"[CLR]
"
```

```
40 BASE=2*4096:POKE 53272,PEEK(53272
)OR 8
50 POKE 53265,PEEK(53265) OR 32
55 IF C>1 THEN 70
60 FOR I=BASE TO BASE+7999:POKE I,0:
NEXT
70 FOR I=1024 TO 2023:POKE I,3:NEXT
75 X=X1:Y=Y1:GOSUB 1000
80 GET X$:IF X$=""THEN 80
85 POKE 650,128
90 IF X$="I"THEN Y=Y-1
100 IF X$="K"THEN Y=Y+1
110 IF X$="J"THEN X=X-1
120 IF X$="L"THEN X=X+1
130 IF X$="B"THEN 2100
140 GOSUB 1000
200 GOTO80
210 END
1000 RO=INT(Y/8)
1100 CH=INT(X/8)
1200 LN=Y AND 7
1300 BI=7-(X AND 7)
1400 BY=BASE+RO*320+CH*8+LN
1500 POKE BY,PEEK(BY)OR(2↑BI)
1600 RETURN
2100 POKE 53265,PEEK(53265)AND 223
```

```
2200 POKE 53272,(PEEK(53272)AND240)O
[illegible]
2300 PRINT"[CLR]":PRINT:PRINT:PRINT:
PRINT:PRINT:PRINT:PRINT
2400 PRINT"AHORA[SPC]ESTA[SPC]EN[SPC]
EL[SPC]PUNTO[SPC]";X,Y:PRINT:PRINT:P
RINT:PRINT:PRINT
2410 PRINT"PARA[SPC]COMENZAR[SPC]CON
[SPC]OTRO[SPC]PUNTO........S":PRINT:
PRINT
2420 PRINT"PARA[SPC]EMPEZAR[SPC]DE[SPC]
NUEVO...............N":PRINT:PRINT
2430 PRINT"PARA[SPC]TERMINAR........
...............F"
2440 GET W$:IF W$=""THEN 2440
2442 IF W$="S"THEN 2500
2444 IF W$="N"THEN 2
2446 IF W$="F"THEN PRINT"[CLR]":END
2448 GO TO 2440
2450 PRINT"INDICAR[SPC]EL[SPC]NUEVO[SPC]
PUNTO[SPC]:"
2500 PRINT"[CLR]":PRINT:PRINT:PRINT:
PRINT:PRINT:PRINT:PRINT:PRINT
2510 PRINT"INDICAR[SPC]EL[SPC]NUEVO[SPC]
PUNTO[SPC]:"
2520 FOR W=0 TO1000:NEXT W
2530 GO TO 5
```



C-64 ó VIC-20

Simon's

Silvia Grandio Luque (12 años)
C/ Castilló, 8. Piso 2º-2ª
Barcelona-21

Supongo que todos conocéis el juego de sonidos y colores "SIMON", pues bien, este juego es una imitación del mismo, pero en lugar de tener que acordarse de colores y sonidos, hay que recordar números.

En la línea 40, podéis cambiar el 1000 por un número más bajo o más alto (cuanto más alto, el número estará más tiempo en pantalla, y cuanto más bajo, lo contrario).

El único fallo que encuentro, es que a los 9 números que te acuerdes no puedes continuar. Yo no lo he podido solucionar, pero te reto a que lo intentes arreglar tú mismo.

```
1 POKE53281,0
3 POKE53280,0
4 PRINT"[CLR]"
5 PRINT"[WHT][CRSRD][14SHIFTC]***SIM
ON***[14SHIFTC]"
10 C=INT(RND(Z)*9)
20 D=10*D+C
30 PRINT"[CRSRD]"D
40 FORX=1TO1000:NEXTX
50 PRINT"[CLR]"
51 PRINT"[CRSRD][14SHIFTC]***SIMON**
*[14SHIFTC]"
60 INPUT"[CRSRD]";N
70 IFN=DTHEN10
80 PRINT"[CRSRD]ERROR,ERA:"D:D=0:INP
UT"[CRSRD]OTRA[SPC]VEZ(S/N)";S$:IFS$
="S"THEN1
```



Invader

VIC-20

Miquel Gras Carreras
Pl. Cots nº 3 - 6º-3.ª
Manresa
(Barcelona)

El juego se trata de ir disparando rayos paralizantes contra un invasor que aparece por la parte superior izquierda con su nave que está situada en una plataforma en el inferior de la pantalla, que se mueve a la derecha con (CRSRD), a la izquierda con (CRSRR), disparando con "Z".

En una sola pasada se puede tocar varias veces debido a que nuestra nave se mueve de dos espacios en dos, mientras que el invasor sólo se mueve de uno en uno, pero ojo con los cuadritos que aparecen a los lados de la plataforma, que no están para decoración, sino que son dos peligrosas bombas que te destruirán.

Sólo queda por decir que este juego no precisa ni ampliación de memoria ni joystick.

```
0 FORSA=1TO2000:NEXTSA:REM MIQUEL GR
AS CARRERAS 1984
1 GOSUB 1000
2 A=8130:TI$="000000"
3 F=8106
4 B=0:SC=0:O=8120:OB=8140
5 POKE36879,8
6 IN=7748
10 FORT=64 TO505
20 PRINT"[HOM][5CRSRR][SPC][RVSON]PU
NTOS[RVSOFF]"SC:PRINT"[HOM][21CRSRD]
[4CRSRR]TIEMPO:[SPC]"TI$
30 POKEO,102
35 POKEOB,102
40 POKEIN+T,81
50 GOSUB310
60 POKEIN+T,32
65 IFT>350THEN1200
70 NEXTT
310 GETA$
315 IFA$="[CRSRD]"THENA=A-2:B=B-2
320 IFA$="[CRSRR]"THENA=A+2:B=B+2
321 IFA$="Z"THENGOSUB5000
325 IFPEEK(A)=PEEK(O)THEN7000
326 IFPEEK(A)=PEEK(OB)THEN7000
330 POKEA,88
335 FORP=1TO150:NEXT
336 POKEA,32
340 RETURN
1000 PRINT"[CLR]"
1010 PRINT"[HOM][25CRSRD][21COMMP]"
1020 FORW=1TO2
1030 PRINT"[21CRSRU]"
1040 NEXTW
1050 RETURN
1200 FORC=10TO25
1205 FORLL=1TO500:NEXT
1206 PRINT"[CLR][11CRSRD][5CRSRR][RVSON]
INVADER":PRINT"[6CRSRD][5CRSRR]PUNTO
S";SC
1210 POKE36879,C
1220 NEXTC
1300 IFSC>10000THEN6000
1400 IFSC<10000THEN6500
2000 END
5000 F=8108:FORKK=1TO15
5005 POKEA,81
5010 POKEF+B,30
5020 FORYY=1TO2:NEXT
5030 POKEF+B,32
5040 F=F-22
5045 IFPEEK(F+B)=PEEK(A)THENSC=SC+23
0
5050 NEXTKK
5060 RETURN
6000 PRINT"[CLR]"
6010 POKE36879,90
6020 PRINT"[3CRSRD][BLK]ERES[SPC]UN[SPC]
GRAN[SPC]TIRADOR[SPC]"
```

```
6025 PRINT"[2CRSRD][2CRSRR][BLK]HAS[SPC]
DERRIBADO[SPC]"SC/230"[5SPC]INVASORE
S"
6026 PRINT"[4CRSRD]EN[SPC]UN[SPC]TIE
MPO[SPC]DE:"TI$
6030 PRINT"[2CRSRD][BLU]QUIERES[SPC]
HACER[SPC]OTRA[SPC]PARTIDA"
6040 INPUTM$
6050 IFM$="SI"THEN1
6060 IFM$="NO"THENEND
6400 END
6500 PRINT"[CLR]"
6510 POKE36879,158
6520 PRINT"[CLR][9CRSRD]ERES[SPC]BAS
TANTE[SPC]MALO,[SPC]SIGUE[SPC]PRACTI
CANDO"
6525 PRINT"[2CRSRD]SOLO[SPC]HAS[SPC]
DERRIBADO"SC/230"[SPC]INVASORES"
6526 PRINT"[3CRSRD]EN[SPC]UN[SPC]TIE
MPO[SPC]DE:"TI$
6530 FORWQ=1TO6000:NEXT
6540 RUN
7000 PRINT"[HOM][9CRSRD][CRSRR]HAS[SPC]
TOCADO[SPC]LA[SPC]BOMBA"
7010 FORZX=22TO31
7015 FORKL=1TO200:NEXT
7020 POKE36879,ZX
7030 NEXTZX:IFSC>10000THENRUN:END
```

VIC-20

Programa quinielas

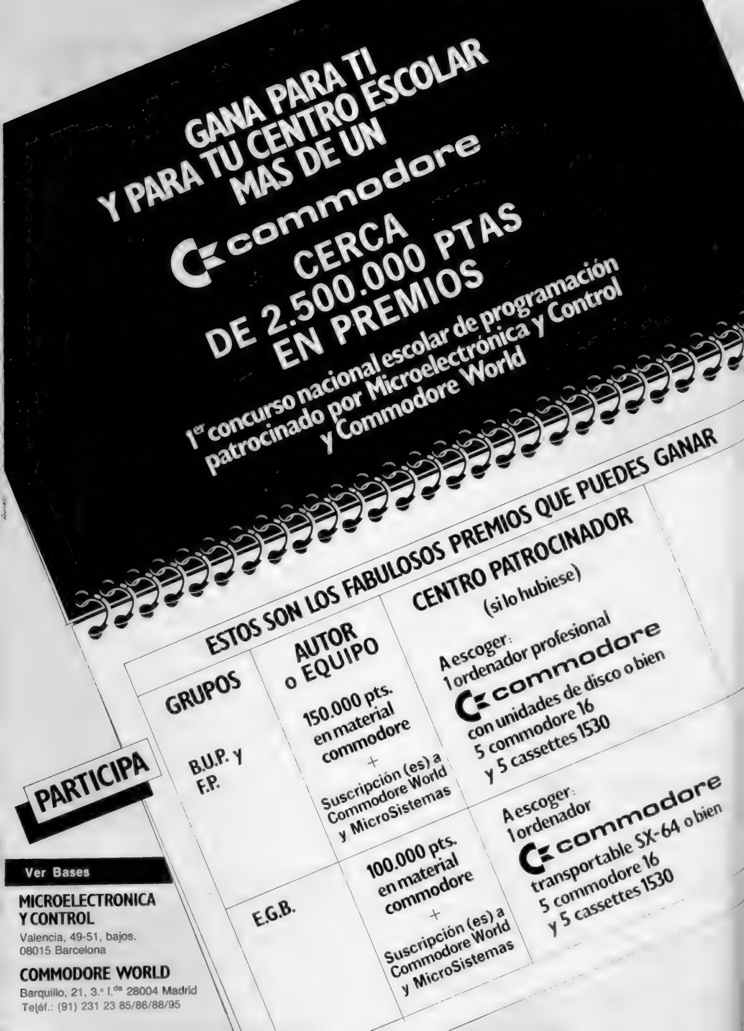
Cirilo Sánchez Barrado
C/ Cedro, 4, Bajo 3
Cornella de Llobregat
(Barcelona)

El programa quinielas juega al azar dándonos las catorce variables, y basta pulsar una tecla, para que nos dé otros catorce resultados.

Al funcionar con un generador de números aleatorios a veces los catorce resultados son idénticos.

La posibilidad de acertar los catorce, con este programa, son las mismas que las de las maquinitas o cualquier procedimiento que juegue al azar.

Para parar el programa pulsar la tecla RUN/STOP.


GANA PARA TI
Y PARA TU CENTRO ESCOLAR
MAS DE UN
commodore
CERCA
DE 2.500.000 PTAS
EN PREMIOS
1er concurso nacional escolar de programación
patrocinado por Microelectrónica y Control
y Commodore World
ESTOS SON LOS FABULOSOS PREMIOS QUE PUEDES GANAR
GRUPOS
AUTOR o EQUIPO
CENTRO PATROCINADOR (si lo hubiese)
B.U.P. y F.P.
150.000 pts. en material commodore + Suscripción (es) a Commodore World y MicroSistemas
A escoger: 1 ordenador profesional commodore con unidades de disco o bien 5 commodore 16 y 5 cassettes 1530
E.G.B.
100.000 pts. en material commodore + Suscripción (es) a Commodore World y MicroSistemas
A escoger: 1 ordenador commodore transportable SX-64 o bien 5 commodore 16 y 5 cassettes 1530
PARTICIPA
Ver Bases
MICROELECTRONICA Y CONTROL
Valencia, 49-51, bajos.
08015 Barcelona
COMMODORE WORLD
Barquillo, 21, 3.º l.da 28004 Madrid
Teléf.: (91) 231 23 85/86/88/95

PROGRAMA QUINIELAS

```
10 POKE36879,8
20 PRINT"[CLR][WHT][4SPC][RVSON][SPC]
QUINIELAS[SPC][2CRSRD]"
30 A=0:B=0:C=0
40 FORX=1TO14
50 Q=INT(5*RND(1))
60 IFQ=4THEN130
70 IFQ=1THEN110
80 A=A+1
90 PRINT"[5SPC]1"
100 GOTO150
110 B=B+1
120 PRINT"[6SPC]X"
125 GOTO150
130 C=C+1
140 PRINT"[7SPC]2"
150 NEXT
160 GETA$:IFA$=""THEN160
170 GOTO20
```



Gráficos 3-D

C-64

David Caballero Valero
Juan Bravo, 40
Telf.: 275 65 52
Madrid-6

En vuestro número 3 (abril del 84) publicasteis un programa super interesante, los gráficos en tres dimensiones para el VIC-20, de Juan Santamaria y Javier Anglada.

Tanto despertó mi interés que raudo fui a probarlo al VIC de un amigo. Quedé sorprendido y desilusionado al comprobar que no funcionaba en el CBM/64.

Así que ni corto ni perezoso me puse a estudiar el programa y ver si existía alguna posibilidad de adaptarlo.

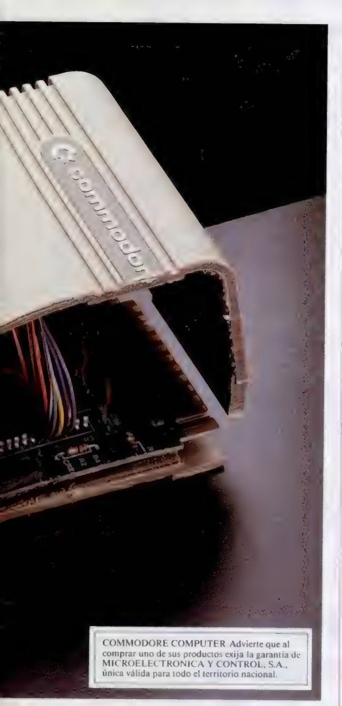
Llegué a la conclusión, tras muchos rodeos, de que había que fabricar una rutina "DRAW" que trazara líneas entre puntos. Así que lo hice y depuré, y lo adapté al programa de Juan y Javi.

El resultado es este. He aprovechado la función de partida. Los parámetros son los mismos que para Grafuics. He introducido además la opción de borrado de alta resolución ("Desactivar s/n"), que lo hace menos pesado en ciertos casos.

Si os quedáis "atascados" en el modo BIT MAP, hacer RUN 700 (y si lo veis muy crudo SYS 8000) y volveréis al modo normal.

El programa es lento. Una vez activado el BIT MAP tarda como un minuto en empezar a trazar los puntos. Y luego tarda bastante en terminar la gráfica.

```
10 REM ********************************
***       *
    *
20 REM *
  *       *
    *
30 REM *   PROGRAMA:GRAFICOS 3-D
  *       *
    *
40 REM *   (ADAPTADO AL CBM/64)
  *       *
    *
50 REM *    POR DAVID CABALLERO
  *       *
    *
60 REM *   Y EL ORDENADOR CBM/64
  *       *
    *
70 REM *    UKR CIRO SOFT 1984
  *       *
    *
80 REM *
  *       *
    *
90 REM ********************************
***
100 PRINT"[CLR]":INPUT"X[SPC]MAX[SPC]
MIN.",X1,X0
110 INPUT"Y[SPC]MAX[SPC]MIN.";Y1,Y0:
INPUT"GIRO";AL:INPUT"FACTOR";Z0
120 INPUT"INTERVALO";I0:DIMM(1,200)
130 INPUT"DESACTIVAR";KOF$:GOSUB370
140 FORJ=0TO102STEP3:Y2=Y0+(Y1-Y0)*J
/102:G=1:FORI=1 TO160
150 X2=X0+(X1-X0)*I/160:GOSUB340:S=(
Z*Z0+J)
160 IFS>200THENS=200
170 IFS<0THENS=0
180 IFD>I0THENG=1:GOTO260
190 IFG=1THENG=0:GOTO240
200 IFS>M(0,I)ANDS<M(1,I)ANDR=0THENR
=1:GOTO320
210 IFS>M(0,I)ANDS<M(1,I)THENGOTO270
```



COMMODORE COMPUTER Advierte que al comprar uno de sus productos exija la garantía de MICROELECTRONICA Y CONTROL, S.A., única válida para todo el territorio nacional.



COMMODORE MPS 801
Impresora de matriz



COMMODORE MCS 801
Impresora cuatro colores



COMMODORE DPS 1101
Impresora de margarita



COMMODORE 1520
Printer plotter cuatro colores

COMMODORE 1541
Unidad de disco 170 K



COMMODORE 1530
Unidad de cassette



COMMODORE 1701
Monitor con color y sonido

JOYSTICKS
PADDLES
SOFTWARE Y LIBROS

10% de su cerebro?

lograr que los textos y gráficos sean más claros y vivos que en un televisor.

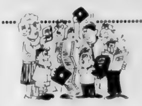
— 3 impresoras y un printer-plotter, para copias en papel, en blanco y negro o color.
— Los joystick y paddles, para conseguir que los juegos sean más excitantes.

Sea exigente. Si su cerebro, el de su COMMODORE 64, puede darle 100, no se conforme con 10.

commodore
COMPUTER
la emoción del futuro

```
220 IFR=1THEN280
230 Q1=(I-1)+50:Q2=I+50:H1=219-(SI+5
0):H2=219-(S+50):GOSUB680
240 IFS>M(1,I)THENM(1,I)=S
250 IFS<M(0,I)THENM(0,I)=S
260 SI=S
270 NEXT:NEXT:GOTO690
280 IFS>=M(1,I)THENSI=M(1,I-1)
290 IFS=<M(0,I)THENSI=M(0,I-1)
300 R=0:GOTO230
310 IFSI<=M(0,I-1)THENS=M(0,I):GOTO2
30
320 IFSI>=M(1,I-1)THENS=M(1,I)
330 GOTO230
340 X=X2*SIN(AL)+Y2*COS(AL):Y=Y2*SIN
(AL)-X2*COS(AL): D=SQR(X*X+Y*Y)
350 Z=EXP(-D*D)*SIN(2*π*D)
360 RETURN
370 REM*SUBRUTINA DE ACTIVACION-----
------------------------------------
-------
380 POKE53280,5:WAQ=8192:POKE53272,P
EEK(53272)OR8:POKE53265,PEEK(53265)O
R32
390 IFKOP$="N"THENGOTO410
400 FORLIK=WAQTOWAQ+7999:POKELIK,0:N
EXTLIK
410 FORGUS=1024TO2023:POKEGUS,5:NEXT
GUS
420 RETURN
430 REM*SUBRUTINA DE COMPARACION----
------------------------------------
-------
440 DH=H2-H1:DQ=Q2-Q1
450 IFABS(DQ)>ABS(DH)THEN490
460 IFABS(DH)>ABS(DQ)THEN550
470 IFABS(DH)=ABS(DQ)THEN600
480 RETURN
490 REM* CASO 1---------------------
------------------------------------
-------
500 IFDQ=0THENH=H1:Q=Q1:GOSUB640:NEX
T:RETURN
510 EQ=DQ/ABS(DQ):PAR=ABS(DH/DQ)
520 IFDH=0THENH=H1:FORQ=Q1TOQ2STEPEQ
:GOSUB640:NEXT:RETURN
530 EH=DH/ABS(DH):H=H1
540 FORQ=Q1TOQ2STEPEQ:H=H+(PAR*EH):G
OSUB640:NEXT:RETURN
550 IFDH=0THENQ=Q1:H=H1:GOSUB640:RET
URN
560 EH=DH/ABS(DH):PAR=ABS(DQ/DH)
570 IFDQ=0THENQ=Q1:FORH=H1TOH2STEPEH
:GOSUB640:NEXT:RETURN
580 EQ=DQ/ABS(DQ):Q=Q1
590 FORH=H1TOH2STEPEH:Q=Q+(PAR*EQ):G
OSUB640:NEXT:RETURN
600 REM* CASO 2---------------------
------------------------------------
-------
610 IFDQ=0THENH=H1:Q=Q1:GOSUB640:RET
URN
620 EQ=DQ/ABS(DQ):EH=DH/ABS(DH):H=H1

630 FORQ=Q1TOQ2STEPEQ:H=H+EH:GOSUB64
0:NEXT:RETURN
640 REM*SUBRUTINA DE TRAZO ---------
------------------------------------
-------
650 HOK=INT(Q/8):SHE=INT(H/8):KAP=HA
ND7:NUN=WAQ+SHE*320+HOK*8+KAP
660 LOK=7-(QAND7):POKENUN,PEEK(NUN)O
R(2↑LOK):RETURN
670 REM*SUBRUTINA DE LINEA ---------
------------------------------------
-------
680 GOSUB430:RETURN
690 PRINT"[CLR]":WAIT(198),1
700 POKE53265,PEEK(53265)AND223:POKE
53272,21:END
870 IFS>M(0,I)ANDS<M(1,I)ANDR=0THENR
=1:GOTO310
```



Control programable de la pantalla

VIC-20

Cirilo Sánchez Barrado
C/ Cedro, 4, Bajo 3
Cornella de Llobregat
(Barcelona)

Los que posean cartuchos de Commodore, habrán comprobado que con el cursor se puede centrar la pantalla. Y los poseedores de algunas cintas en código máquina, que la pantalla se puede desplazar en vertical y horizontal.

Por ejemplo, el que posea el programa centipede habrá visto que la pantalla desaparece por completo y aparece por la parte interior del televisor hasta colocarse en su sitio.

Como esto solo lo podía encontrar en programas en código máquina, empecé a experimentar con los Pokes hasta que dí con el Poke 36897 y pude comprobar que podía controlar la posición de la pantalla en vertical. Al teclear Poke 36897,1, la pantalla apareció en la parte superior del televisor, al ir ascendiendo el número la posición de la pantalla iba bajando. Al teclear Poke 36897,200, la pantalla desapareció totalmente.

Ya que con este Poke sólo podía controlar la vertical, descendí al Poke 36896 y pude controlar la posición de la pantalla en horizontal. Al teclear Poke 36896,1, la pantalla se posicionó a la izquierda. Ascendiendo el número, la pantalla se desplazaba a la derecha. Al llegar al Poke 36896,24, la pantalla en posición derecha se llenó de símbolos y letras; al ascender más el número, la única forma de recuperar la pantalla en su sitio y limpiarla, fue apagar el Vic.

Al bajar el Poke 36895, comprobé que hacía lo mismo que el Poke 336879, controlar el color de la pantalla y el del marco.

El programa es una demostración de cómo controlar estos Pokes. Al principio controla la pantalla en vertical subiéndola y bajándola hasta desaparecer, la segunda parte del programa nos controla la posición de la pantalla con los cursores (CRSR-D) y (CRSR-R).

10- 20 Cambia el color del fondo de la pantalla, a negro y la pantalla desaparece.
60- 70 Sube la pantalla a su posición normal.

80 Detecta la pulsación de una tecla.
90-100 Baja la pantalla hasta que desaparece.
110-130 Cambia el contenido de la pantalla.
140-150 Vuelve a subirnos la pantalla a su posición.
160 Para el programa durante unos diez minutos.
170-180 Desplaza la pantalla a la parte superior izquierda del televisor, para poderla central con los cursores.
180 Da el control horizontal, desde la posición 1 a la posición 20 con el cursor (CRSRR), para desplazar la pantalla a la derecha. Al llegar a la posición 20 nos vuelve, la pantalla a la posición 1.
190 Da el control vertical de la posición 10 a la 60, para el cursor (CRSRD), y poder bajar la pantalla. Al llegar a la posición 60 nos la vuelve a poner en la posición 10.

```
10 POKE36895,24:V=36897:A=200:PRINT"
[CLR][CRSRD][RVSON][SPC]CONTROL[2SPC]
PROGRAMABLE[SPC]"
20 POKEV,A:PRINT"[4SPC][RVSON]DE[SPC]
LA[SPC]PANTALLA":PRINT"[2CRSRD][2SPC]
MOVIMIENTO[SPC]VERTICAL[CRSRD][3SPC]
DE[SPC]LA[SPC]PANTALLA"
30 PRINT"[2CRSRD][BLU][2SPC]DEL[SPC]
[RED]POKE36897,38":PRINT"[CRSRD][BLU]
[2SPC]AL[SPC][RED]POKE36897,200[BLU]
"
40 PRINT"[3CRSRD][3SPC][RVSON]PULSA[SPC]
UNA[SPC]TECLA[RVSOFF]"
60 POKEV,A:A=A-0.5:IFA<38THEN80
70 GOTO60
80 GETA$:IFA$=""THEN80
90 POKEV,A:A=A+0.5:IFA>200THEN110
100 GOTO90
110 PRINT"[CLR][4CRSRD][3SPC]MOVIMIE
NTO[SPC]DE[SPC]LA":PRINT"[CRSRD][6SPC]
PANTALLA[SPC]EN"
120 PRINT"[CRSRD][SPC]VERTICAL[SPC]P
OKE[SPC]36897":PRINT"[CRSRD][SPC]HOR
IZONTAL[SPC]POKE36896"
130 PRINT"[2CRSRD][PUR][2SPC]CENTRE[SPC]
LA[SPC]PANTALLA":PRINT"[CRSRD][4SPC]
CON[SPC]EL[SPC]CURSOR[BLU]"
140 POKEV,A:A=A-0.5:IFA<38THEN160
150 GOTO140
160 FORZ=1TO10000:NEXT:PRINT"[2CRSRD]
PARA[SPC]PARAR[SPC]EL[SPC]PROGRAMA[CRSRD]
[4SPC][RVSON]PULSA[SPC]ESPACIO"
170 H=36896:B=1:A=10
180 POKEH,B:GETB$:IFB$="[CRSRR]"THEN
B=B+0.5:IFB>20THENB=1
190 POKEV,A:IFB$="[CRSRD]"THENA=A+0.
5:IFA>60THENA=10
200 GOTO180
```

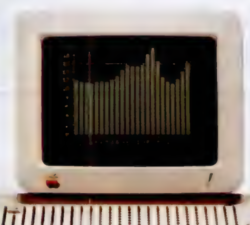
Todos los meses publicaremos programas para los Commodores.



MicroSistemas

y le llevaremos a su casa un APPLE II c

Suscríbase a MicroSistemas antes del 30 de noviembre próximo y participará en el sorteo de un magnífico ordenador personal APPLE II c.
Para los que ya estén suscritos también tendrán opción de conseguir el APPLE II c. Si desea conocer su número en el sorteo llámenos al teléfono 231 23 95.
Habrá también otros premios para nuestros suscriptores que daremos oportunamente a conocer. El sorteo tendrá lugar en nuestro stand del SIMO.



BOLETIN DE SUSCRIPCION A MICROSISTEMAS

Nombre y Apellidos ....
Puesto ....
Compañía ....
Actividad Empresa ....
Dirección .... Teléf.: ....
Ciudad .... D. P. .... Provincia ....

Suscripción anual 2.950 ptas.
Deseo iniciar la suscripción con el próximo número
☐ Adjunto cheque por 2.950 ptas.

☐ Reembolso más gastos del mismo al recibir el primer número de la suscripción.
(Enviar a Gravina, 13-1º Izda. Madrid-4 - Teléf.: 231 23 88/95)

Laboratorios de informática en la enseñanza

Muchos son los colegios que nos están llamando pidiéndonos información sobre la compra de ordenadores, instalación de un laboratorio, preparación del profesorado, etc.

Por fin creemos poder dar un solución y un lugar a donde los colegios pueden dirigirse exponiendo las necesidades del centro y esperando un asesoramiento lógico y profesional.

La firma Informática y Educación S.A es una empresa dedicada exclusivamente a profundizar en el estudio de las necesidades de la enseñanza de la Informática y campos asociados en el área de la educación.

No es necesario que los colegios se planteen un alto dispendio para comenzar la enseñanza de esta materia.

Desde el VIC al 64, pasando por el nuevo 16, se puede comenzar con una modesta instalación e ir aumentándola y sofisticando según la vayan haciendo las necesidades del centro.

Teniendo en cuenta que cada ordenador es utilizado al tiempo por equipos de tres alumnos (4 máximo), 5 ordenadores es un número suficiente para cubrir las necesidades de un laboratorio con clases de 15 a 20 alumnos.

A partir de aquí, y con este requerimiento mínimo, Informática y Educación ha desarrollado todo tipo de ampliación y sofisticación que el centro requiera.

Uno de sus principales objetivos fue salvar ese escollo de difícil solución, ignorado consecuentemente por muchos instaladores de equipos informáticos, que es el aislamiento del alumno frente a la máquina.

Así nació la ED-NET (red de área local para la educación) que consiste en comunicar los diversos ordenadores entre sí, todos los ordenadores a un único periférico —impresora, unidad de discos, etc.— y todas las unidades manejadas por los alumnos a la unidad manejada por el profesor. Cada ordenador puede funcionar independientemente, en grupos o en clase general dirigida por el profesor a través de su unidad.

Esto lo comprobamos en la práctica visitando el laboratorio instalado con este sistema en las escuelas Padre Piquer donde imparten las enseñanzas de E.G.B., F.P. 1º y 2º grado y B.U.P.



Colegio Lorenzo. El ordenador sobre la mesa redonda es la unidad central del profesor.

Aquí pudimos ver la eficacia y profesionalidad de esta instalación en un laboratorio montado con 14 puestos de trabajo en los que se encontraban trabajando 40 alumnos.

Durante esta breve visita pudimos comprobar los siguientes puntos:

El profesor puede comprobar, en cualquier momento, y sin necesidad de "estar encima" de los chicos, qué está sucediendo en cualquiera de las pantallas. Esto significa que desde su unidad el profesor puede "entrar" en el ordenador que desee a fin de corregir errores, hacer recomendaciones, corregir un trabajo, etc.

Igualmente y de la misma forma, el profesor puede "enviar" su propia pantalla al puesto deseado como ayuda a ese puesto, o bien puede enviarla simultáneamente a todos los puestos para una explicación general.

Esto puede realizarlo sin "borrar" el trabajo que los alumnos estén realizando ya que éste queda depositado en la memoria y el alumno puede volver a "llamarlo" donde lo dejó una vez que el profesor le dé vía libre.

Un punto importante es que la unidad de profesor no es un gran y caro ordenador central, sino un aparato más, idéntico al de los alumnos, en otras palabras, un VIC, un 16 ó un 64, en este caso de las escuelas Piquer, un 64.

El paquete de Soft, que entregaron a Padre Piquer con los ordenadores, y que hemos visto funcionar, es muy completo, incluyendo los lenguajes Logo, Pilot, Forth, Assembler, Simon's, Power, Graphics, Superbasic, Base de Datos, proceso de Textos y por supuesto una colección bastante completa de programas educativos, académicos y juegos.

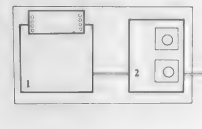
Este mismo paquete, que hemos visto en castellano, lo tienen preparado también en euzkera y en catalán.

Según nos cuenta Ramón Leonato, directivo de Informática y Educación, tienen ya preparada otra serie de productos ▶

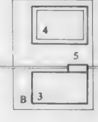
RED ED-NET PARA N PUESTOS

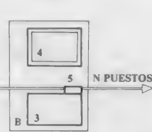
1. **Impresora**
2. **Discos**
3. **Commodore 64**
4. **Monitor**
5. **Dispositivo red**

A. **Puesto master**
B. **Puesto normal**











Laboratorios de las escuelas Padre Piquer.

pensados y diseñados especialmente para la educación, tales como un equipo para control automático de secuencias para iniciar a los estudiantes en la robótica; una pantalla digital que podrá ser utilizada para iniciar en el "mágico" mundo del ordenador a los chiquilines ya que el ordenador responde a las órdenes por simple tacto del dedo; un paquete con el sugestivo nombre de "Mágico", paquete especialmente diseñado para la enseñanza asistida por ordenador, de cualquier materia académica, con el que el profesor, según nos dicen, podrá, desde llevar toda una clase según sus propios métodos y criterios, hasta la gestión completa del centro...

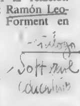
Repetimos que aunque presumimos que estos productos serán tan interesantes como suenan, no hemos tenido todavía la oportunidad de comprobarlos.

Según nos dicen, estarán disponibles entre los proximos 15 días y un mes, lo que esperamos poder comentaros sobre el tema en el próximo número.

Queremos dar las gracias desde aquí a José María Picazo, jefe de Estudios de las Escuelas Padre Piquer por toda la colaboración que nos prestó así como a José Luis Vicente, profesor de informática, que aguantó con una gran amabilidad y una sonrisa la irrupción que hicimos en pleno periodo de clase, sin olvidar a los alumnos de primer curso de 2º grado que soportaron de lo más educadamente las "entradas", interrupciones y "espionajes" a los que les sometimos.

Las personas que llevan la relación directa con los colegios son: Ramón Leonato en Madrid y Juan Forment en Barcelona.

Barcelona.
Joaquin Folguera, 2
Teléfono 247 40 03



Madrid
General Pardiñas, 74 - 3º
Teléfono 401 28 48

Albareda: órgano de verdad

En Sonimag 84 tuvimos la oportunidad de ver también los últimos accesorios lanzados al mercado para el C-64, especialmente destinados a explotar todas sus posibilidades musicales.



Muchos nos habíais escrito pidiéndonos señas de casas que comercializaran teclados tipo órgano, adaptables al C-64, pero no pudimos contestaros ninguna en concreto, pues bien, ahora en España la casa Albareda de Lérida ha puesto a la venta un teclado con un cartucho que contiene el interface y el programa para utilizarlo como mono o polifónico.

Para utilizarlo se necesita un C-64, un monitor o televisor, y una unidad de disco o cassettes, ya que el programa está disponible tanto en cinta como en disco.

El teclado es del tipo profesional con cuatro octavas completas de Do a Do, no es una de esas "cosas" que acaban de lanzar en Estados Unidos para superponerlas al teclado normal del C-64, es un teclado totalmente independiente y sólo está unido al ordenador por un cable plano de buena calidad.

Toda la circuitería electrónica del interface se encuentra distribuida entre el cartucho que se conecta al port de expansión y al propio teclado, eliminando de este modo los problemas que pudieran originar las "cajas colgantes" en cualquier sitio. Por tanto se ha cuidado el diseño para evitar averías, y el cartucho es de los pequeños, similar a los de juegos, por lo que no estorba detrás del ordenador.

Con el teclado suministran un par de programas, bien en cinta o en disco, que permiten utilizar el C-64 con el teclado como sintetizador monofónico o como órgano polifónico. Como sintetizador monofónico podemos escoger entre 13 instrumentos: trompeta, brass, clarinete, campanas, flauta, guitarra, wha-brass, instrumentos de cuerda, piano, dos tipos de órganos eléctricos, acordeón, o sintetizador. Cada uno de estos instrumentos se puede ajustar para distintos matices por medio de una de las opciones del programa, por ejemplo podemos hacer que la trompeta suene con sordina o sin ella, dentro de los de cuerda obtener toda una variedad de instrumentos, etc.

Si lo utilizamos como teclado de órgano polifónico, podemos obtener otros 5 instrumentos: spinete, acordeón, campanas tubulares, flauta o banjo. Y también por medio de las teclas de función variar los parámetros de cada instrumento (ataque, declive, sostenimiento, relajación, volumen, octavas trabajadas, forma de onda, y ciclo de operación).

Creo que el lanzamiento de este nuevo producto hará las delicias de muchos aficionados y compositores que lo estaban esperando, su precio de lanzamiento era unas 37.000 pesetas. Si deseáis más información podéis dirigiros a:

Albareda, instrumentos musicales y accesorios
Carmen, 19
Tárrega (Lérida)

NOVEDADES SONIMAG

Idealogic: Software preescolar y primer ciclo de EGB

Idealogic de Barcelona ha lanzado de Software didáctico dirigido a los chicos de corta edad. Han preparado una selección de los productos de las mejores marcas americanas y lo han adaptado al español para los ordenadores más populares de nuestro mercado (Apple IIe, Commodore 64 y Spectrum). El equipo de Idealogic lo forman educadores, psicólogos, programadores, etc., todos son grandes profesionales en sus campos y el resultado de su unión ha sido la buena selección de programas educativos que han preparado.

Los títulos que tuve oportunidad de ver en Sonimag y seguir probando en la oficina, son:

—Compulandia, programa destinado a niños de edades entre 3 y 10 años, es una colección de juegos que incluyen reconocimiento de letras, crear dibujos, completar series numéricas y de figuras, dar animación y sonido a las palabras y nombres, etc... Entre sus beneficios educativos podemos citar: mejorar la capacidad de lectura del niño y el nivel de escritura, ayudar al niño a aprender a contar, desarrollar la creatividad del chico y familiarizarle con el teclado y el ordenador.

—Mil Caras, para edades de 3 a 10 años, son tres juegos en uno, uno de ellos consiste en ponerle los rasgos a una cara que originalmente está en blanco. En el segundo, por medio de la cara creada, el niño puede darle órdenes de reír, llorar, guiñar un ojo, etc., estimulando la expresividad del chico. El tercer juego es de memoria, la cara va realizando una serie de gestos, y el chico tiene que repetirlos desde el teclado. Entre sus beneficios está el ayudar al niño a entender y utilizar conceptos fundamentales de la informática como menús, gráficos, cursor, tecla return, etc. Estimula la creatividad, concentración y memoria, además de introducir al chico en los conceptos de codificación y abstracción.

—Teclas Divertidas es para chicos de 3 a 11 años, son otros tres juegos, uno es de reconocimiento de letras que van cayendo

desde la parte alta de la pantalla. Otro consiste en teclear la palabra que identifica al objeto que cae, y el tercero en relacionar mediante números las diversas palabras y figuras que aparecen en la pantalla. Sus principales beneficios son el familiarizar al chico con el teclado y el reconocimiento de letras y números.

—ALF en las Grutas de Color es para los más pequeños de la casa (de 3 a 8 años), el protagonista es un simpático bichejo que baila al son de la música, es muy fácil de jugar, son unos túneles de colores y formas diferentes que llevan asociados distintos sonidos, y estimulan al chico en gran manera. Sus beneficios son desarrollar la planificación estratégica, mejorar la comprensión de la relación causa-efecto e introducir al chico en la utilización del ordenador.

—El Rancho es para los mayores (de 6 a 12 años), permite al chico crear escenas en la pantalla, darles animación y música. Dispone de una amplia gama de personajes y figuras para colocar en la pantalla moviéndolas gracias al joystick, y las escenas pueden guardarse en cassette.



Este juego familiariza al chico con las funciones elementales del ordenador (menús, pantallas, etc.), potencia el desarrollo de diseños y composición de gráficos, la creatividad y la capacidad de planificación del trabajo a realizar.

Todos los programas vienen acompañados de un manual, poco usual en las otras casas, en el que da una serie de consejos a los padres, unas ideas para los niños, expone la finalidad educativa del juego y sus beneficios. Las instrucciones que acompañan al manual vienen claras y concisas, y da gusto ver una alternativa a la gran "invasión de invasores" que están sufriendo los pequeños de la casa y que sólo estimula la agresividad. Creo que son unos juegos interesantes y bien desarrollados tanto en el aspecto del propio programa como en su valor educativo.

Los programas vienen en cintas, y sus precios se han unificado: 2.400 pesetas cada uno.

Las señas de Idealogic son:

Gran Vía de Carlos III, 97 K,
08028 Barcelona
Teléfono (93) 330 33 08

Catálogo Software

Próximamente nos sorprenderán con más lanzamientos que tuve oportunidad de ver en la fase de preparación. Entre ellos ya tienen preparado un nuevo Joystick para los equipos Commodore (VIC y C-64), que permite un control más suave, también una tabla gráfica para digitalizar mapas o dibujos, unos nuevos monitores de alta resolución en color con sonido y un precio relativamente asequible (unas 60.000 pesetas), nuevos juegos del tipo de aventuras, programas de dietética, aerobic, etc...

Ferre Moret: Soft y Hard

Vimos el nuevo juego que esta casa presentó en el salón, y hemos de reconocer que la reproducción de la pantalla que sacamos en color en la revista no puede igualar jamás a la realidad, la primera impresión al ver la pantalla del juego es que nos encontramos volando sobre Washington, podemos distinguir la Casa Blanca, el Capitolio, el Washington Memorial, toda la avenida, las casas e incluso



una bandera agitada por el viento. El mérito de este programa reside precisamente en la calidad de los gráficos que han conseguido, son una perfecta fotografía como pudisteis comprobar los que pasateis por Sonimag y lo visteis en su stand. También hay que decir que el final del juego es aún mejor, si consigues pasar de 150 puntos, pasas a luchar contra la nave madre en el espacio, toda esta parte del juego está realizada con gráficos de alta resolución tridimensionales, y al ir moviéndose hacia un lado u otro da la sensación de viajar entre las estrellas a una gran velocidad. Por fin si consigues destruir la nave nodriza, te reciben en Washington con una gran fiesta de fuegos artificiales y música. Creo que es un juego digno de verse, y si te gusta, cómpralo. Su precio es un poco alto (9.975 pesetas), pero es el primer juego que veo con esa calidad de gráficos.

También tuvimos oportunidad de comprobar el funcionamiento de las tarjetas de 64K RAM para el VIC-20, sobre las que me habíais preguntado en bastantes cartas, pues bien, ahora puedo contestar: en primer lugar la RAM disponible para programas y datos en todo momento son sólo 28K en la zona de programas Basic y 3K entre las direcciones 1024 y 4096 y 6K en las direcciones que normalmente se utilizan para los cartuchos de juego y el super-expander. De este modo podemos tener siempre enchufada la tarjeta y cuando queremos cargar un juego, el Forth o el super-expander sólo tenemos que cargarlo de cinta, con lo que no estropeamos el conector de expansión al enchufar y desenchufar los cartuchos constantemente.

El sistema para utilizar el resto de la memoria que queda oculta bajo la ROM y las unidades de entrada salida del VIC-20, consiste en utilizar los comandos LOAD y SAVE con el número de periférico comprendido entre 200 y 207 (el cassette es el 1, el disco el 8, etc.). De este modo da la sensación de disponer de 8 cassettes o discos super-rápidos que en menos de un segundo te cargan en la memoria los programas que tenías ocultos, y que se mantienen guardados hasta que apagues el VIC o guardes otro en su mismo lugar. Naturalmente el contenido de estos ocho cassettes ocultos se puede transferir a cinta o disco par utilizarlo en otra ocasión.

La casa Ferre Moret tiene además unos programas de contabilidad, control de Stock, etc., ya preparados para utilizar todas las posibilidades de esta tarjeta y de las de 40/80 columnas.

Otra novedad que presentó en la feria era la tarjeta de 80 columnas para el C-64, similar en funcionamiento y calidad a la del VIC 20, y al mismo tiempo anunció la próxima presentación de una tarjeta de 256K RAM para el C-64, que imaginamos funcionará de modo similar al de la de 64K del VIC, y algunos juegos más que tiene en cartera para comercializar próximamente.

Las señas de Ferre-Moret, S. A. son:

Tusset, 8, Entresuelo 2
08006 Barcelona
Teléfono (93) 218 02 93

☆ ☆ ☆ ☆

Magia



La MAGIA son trucos, la MAGIA es divertida.

La MAGIA es hacer lo que nadie se ha atrevido y resulta ser la fuente más completa de información para la informática práctica.

La MAGIA es una sección mensual llena de consejos, trucos, de esto y aquello del mundo del software, hardware y aplicaciones, trucos descubiertos por los demás que hacen que la informática sea más fácil, más divertida o más animada.

MAGIA habla de ideas sencillas, programas de una sola línea, subrutinas útiles, hechos de informática poco conocidos y otras cosas de interés.

Los trucos de magia enviárnoslos comprobados, pues hay varios incorrectos. ¡Ah! y no nos mandéis trucos repetidos... ¡Listos!. Premios: Ver pág. 5.

PETS Y VELOCIDAD

Amigos de Commodore:

Os envío un par de trucos para que los publiquéis (si podéis) en la sección de MAGIA. El primero sirve para detener un programa interactivo, pulsando la tecla SHIFT LOCK:

WAIT 152, 1,1 - CBM 2000, 3000, 4000 y 8000 con el basic 2 ó 4
WAIT 516, 1,1 - CBM 2000 con el basic 1
WAIT 653, 1,1 - CBM 64 y VIC 20

Lo que hace esta sentencia es comprobar si la tecla SHIFT (o SHIFT LOCK) está pulsada: si la respuesta es negativa el control pasa a la siguiente sentencia; si es afirmativa, espera hasta que la tecla deje de estar pulsada.

El segundo truco sirve para los ordenadores 2000 y 3000
POKE 59468, 62

Esta sentencia aumenta considerablemente la velocidad de edición en pantalla, alcanzando la velocidad de la serie 4000. En mi ordenador (un 2000) la pantalla sufre varias perturbaciones antes de restablecerse la normalidad, no ocurre así con la serie 3000.

Marco Rossi jr.
C/Masferrer nº 14
08028 Barcelona

NOTAS: En los equipos CBM 2000, 3000, 4000 y 8000 no hemos podido comprobar este truco (el primero)

El segundo truco apareció hace unos años (dos o tres) en la revista inglesa Computing Today, y funcionaba, pero unos meses más tarde recibieron algunas cargas que publicaron, quejándose de que los PETs se averiaban por culpa del POKE. Al parecer utilizar algunas veces este POKE no era malo, pero el uso continuado perjudicaba de algún modo a la unidad de pantalla (no lo podemos asegurar, sólo lo mencionamos, por si acaso).

BUFFER DE TECLADO VIC-20 Y C-64

He leído en varios artículos de diversas revistas, la utilización de la posición de memoria 198, y en otros además utilizaban también las posiciones 631 a 640, pero no he encontrado ningún artículo con una explicación. Hace poco tiempo que leí que estas posiciones son el buffer del teclado.

La explicación del funcionamiento del buffer parece que se intenta mantener en secreto. Por lo que sé, durante la ejecución de un programa en basic, el VIC-20 escrudriña el teclado sesenta veces por segundo. Si se pulsa una tecla, el ordenador no la interpreta sino que la almacena en el buffer a partir de la posición 631 y hacia la 640. La posición 198 sirve para contar el número de teclas pulsadas.

Una vez que se detiene el programa (mediante un END, INPUT, GET$ o STOP), el buffer se vacía de nuevo, imprimiendo las teclas u órdenes pulsadas.

Para entenderlo mejor y que sirva de ejemplo, podríamos aprovechar esta característica del VIC-20 para meter una función (especialmente interesante para gráficos) mediante un INPUT, con el cual haremos que un programa se escriba una línea el mismo.

Tomo de ejemplo el programa de gráficos en tres dimensiones que me publicasteis en el número 3 de vuestra revista.

Lo que vamos a hacer va a ser llenar el buffer de teclado desde el mismo programa que vamos a modificar. Para vaciar el buffer pondremos un END, luego haremos un "return" sobre la línea a introducir que antes habremos escrito en pantalla y por último haremos un RUN a la siguiente parte del programa.

Por último recordar que estamos cambiando una línea de programa y por tanto se borran todas las variables. El mismo sistema se puede utilizar para mandar LOADs, SAVEs, NEWs, LOAD + RUNs... y todo lo que se os ocurra; sólo es cuestión de llenar el buffer y parar el programa.

```
10 INPUT "Función f(x,y)="; a$
20 PRINT "[SHIFT CLR] 500; z="; a$
```

```
30 POKE 631, ASC("[CLR]"): POKE 632, 13: POKE 633, ASC("R")
40 POKE 634, ASC("[SHIFT U]): POKE 635, ASC("1"): POKE 636, ASC("0");
50 POKE 637, ASC("0"): POKE 638, 13
60 POKE 198,8
70 END
```

Alberto y Javier Anglada Cantarell
Teléfono: 247 91 62
c/ Mandri, 68
Barcelona

Club de ordenador personal EINEA
Teléfono: 215 72 74
Rbla. Catalunya, 50
Barcelona

☆ ☆ ☆ ☆ ☆

IMPRESORA Y BUSQUEDA DE PROGRAMAS

Estimados amigos de Commodore World:

Me llamo Román Estébanez y vivo en Pamplona. Poseo un Commodore 64, un cassette y una impresora MPS-801. Os mando estos trucos que espero sean de vuestro agrado.

Son los siguientes:

1. Este truco facilita el paso en la impresora del modo mayúsculas y gráficos al de minúsculas. Consiste en pulsar dentro del Print# la tecla de cursor hacia abajo. En la pantalla aparecerá una Q en vídeo inverso. Esto hará que la impresora comience a escribir en modo mayúsculas sin necesidad de usar ningún CHR#

2. Se trata de un truco para facilitar la búsqueda de un programa en el cassette. Basta con colocar un cable desde la patilla N5 de la salida audio/vídeo al cable n4-d (blanco-lectura del cassette) en el Port del cassette.

Abriendo el volumen con un POKE 54296, 1 podremos monitorear a través del altavoz de nuestro televisor lo que hay grabado en la cinta, ayudándonos a encontrar los programas por su pitido inicial.

NOTA: Este truco es muy sencillo en el cassette antiguo, ya que basta con introducir un cable pelado por el hueco que deja el cable blanco en la parte trasera del conector. En el moderno habrá que desmontar la clavija y hacer la conexión oportuna.

Román Estébanez Santesteban
C/San Fermín, 49 izq.
Pamplona (Navarra)
Teléf.: (948) 23 63 72

☆ ☆ ☆ ☆ ☆

PRINT AT C-64

Y ahí va el truco. Se trata de una rutina en código máquina para simular PRINT AT (ya sé que no soy muy original, perdonadme). Utiliza la rutina del kernal PLOT. El programa está hecho en un C-64. Para usarlo, después de ejecutar el programa, se escribe SYS 49300. POSX, POSY, "MENSAJE" cuando quieras escribir algo.

```
10 REM **PRINT AT**
20 SU = 0: I = 49300
30 FOR X=0 TO 30: READ A: POKE I+X,A: SU=SU+A: NEXT
40 IF SU<>4300 THEN PRINT "ERROR EN LINEAS DATA":
   END
50 PRINT "UTILIZA SYS 49300, POSX, POSY,"; CHR$(34);
   "MENSAJE"; CHR$(34)
60 END
70 DATA 32,241, 183, 224, 40, 176, 9, 134, 252, 32, 241, 183, 224, 24
80 DATA 144, 3, 76, 72, 178, 24, 184, 252, 32, 240, 255, 32, 253, 174
90 DATA 76, 160 ,170
```

Javier Ontañón Ruiz
Estrella Polar, 22, 12 B
28007 Madrid

☆ ☆ ☆ ☆ ☆

ACELERADOR EN EL C-64

¡¡Aquí estoy, tal como había dicho!!

Bueno, aquí están unas MAGIAS que no son las que os había dicho que os iba a mandar, pero es lo mismo.

La primera, es un acelerador para el C-64:

Tecleando POKE 56325, 0 los programas se aceleran el cursor se retarda.

Tecleando POKE 56325, 255 los programas se retardan el cursor se acelera, y se mueve como un Fórmula-1.

Tecleando POKE 56325, 51 todo vuelve a la normalidad.

OJO, con estos POKEs, el reloj deja de funcionar correctamente. No os fiéis.

La segunda MAGIA es para controlar el color de las letras y símbolos gráficos de éste:

POKE 646, color

¿A que soy un genio?

P.D.: ¡¡como encuentre al tío que dice en el Manual del Usuario eso de 'no necesita saber música para tocar algo con su COMMODORE-64' lo mato!!.

El único sonido que me sale a mí es el que hago cuando pulso las teclas o cuando se pone la DATASSETTE en marcha.

Juan Rafael Oscar Martín y Mihalic
Guayadeque, 4, 3º Izda.
35009 Las Palmas

☆ ☆ ☆ ☆ ☆

BINGO

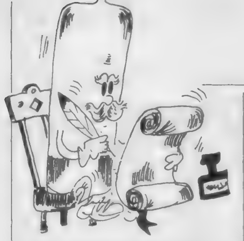
Esta es la primera vez que escribo y me gustaría que publicaseis en la sección de MAGIA este sencillo BINGO que, creo, hará las delicias para los que empiezan, que no están para copiar largos listados. Como veréis, está basado en la función RND y una vez copiado el listado y pulsado RUN, saldrá el primer número y tan sólo hará falta pulsar RETURN para que los números vayan apareciendo ¿fácil no...?

Vale para el C-64 y el VIC-20.

LISTADO DEL "BINGO" PARA MAGIA.

```
10 PRINT "[CLR]"
20 PRINT "==BINGO=="
30 REM PEDRO CARRERAS
40 PRINT ""
50 PRINT INT (RND (A)*99)
60 INPUT
70 RUN
```

Pedro Carreras Luque
c/ Ampurdá, 30-2º-4.ª
Parets del Vallés
Barcelona



CARTA BLANCA

C-64 MUSICAL CON MIDI

Os pediría me indicarais direcciones de casas de software (nacionales o extranjeras) que comercialicen aplicaciones musicales para el C-64, desarrollos para el COMMODORE y el M.I.D.I (Musical Instrument Digital Interface).

DANIEL RIOS ARANDA
C/ROGER DE LA FLOR, 5 - 3º A ESC. DCHA. - 50010 ZARAGOZA

Puedes ponerte en contacto con la casa Vietronic, S.A., que es la que acaba de lanzar en el mercado español el M.C.I. (Midi Computer Interface) para el C-64 y el Spectrum 48K (mira en noticias Sonimag). Las señas de esta casa son: Vietronic, S.A. C/Bolivia, 239, 08020 Barcelona. Tel. 93-3074712.

OPERADORES LOGICOS

Tengo algunas preguntas. 1) ¿Cómo se utilizan y qué indican los operadores lógicos ANDy OR? 2) ¿En un registro cómo puedo poner a cero o a uno determinados bits según mi voluntad? (lo digo por el registro 17 del Chip de vídeo del C-64). 3) ¿Qué es el Bit-map? 4) ¿Cómo funciona (y qué es) la instrucción SYS? Como habréis visto casi todas mis dudas van de cara a la alta resolución de la que no tengo ni idea de cómo se trabaja con ella, por lo que propongo que dediquéis algún artículo al respecto.

CESAR VILLELLA BONA
C/PERE, 53, 3ª, 1ª
BARCELONA

1.—El operador AND hace que el resultado de la operación sea verdad (uno) sólo si todas las premisas son verdad. El resultado de la utilización de OR es verdad cuando cualquiera de las premisas es verdad. Esto es aplicable a los bytes considerando que las operaciones se realizan bit a bit (el bit cero de una de las premisas con el cero de la otra da el bit cero del resultado, etc.).

2.—Para poner a cero un bit determinado puedes utilizar el AND, y para ponerlo a uno el OR. Ejemplo: A=A OR 16 pone a uno el bit cuarto de A; A=A AND 253 pone a cero el bit primero de A.

3.—El bit-map es uno de los sistemas de representación en pantalla que utilizan los ordenadores; hace que cada punto de la pantalla se corresponda con un bit en la memoria, de modo que para iluminar el punto se pone el bit a 1, y para apagarlo se pone a cero.

4.—La instrucción SYS hace que la CPU salte a un programa en código máquina que se encuentre colocado en la dirección que indica el número que sigue a SYS.

VIC-20 RADIO ACTIVO

Me gustaria que me respondieran a dos preguntas: ¿Existe alguna aplicación de mi ordenador personal (VIC-20) en CW o RTTY? ¿Cuál es esa aplicación?

JOSE BRAVO RODRIGUEZ
CORDOBA

En algunos de los distribuidores de Commodore (grandes almacenes por ejemplo), he podido ver algunos interfaces y programas para utilizar el VIC-20 en CW y RTTY, el nombre comercial que le dieron al más popular, fue VICHAM (Ham-radio amateur), y también tenías programas de Micro Electrónica y Control, para la gestión de las tarjetas QSL's etc...

PROBLEMAS CON EL 64

Tengo un Commodore 64 y desde el principio, he observado que con frecuencia al comenzar a trabajar, el funcionamiento es incorrecto, apareciendo a veces el mensaje inicial en forma estraña o el mensaje: OUT OF MEMORY IN Ø, igualmente mezclado con signos diversos.

El problema se corrige desconectando y conectando de nuevo el ordenador una o más veces.

Si existe un cartucho conectado, como SIMONS'BASIC o FUTBOL, sobre todo con este último, puede ser necesario desconectar y conectar hasta cinco o seis veces para que el funcionamiento (no sólo en mensaje inicial) sea correcto.

Este fenómeno es menos acusado si el ordenador ha permanecido cierto tiempo en servicio.

Quisiera saber si se trata de un defecto o avería o es algo que suele ocurrir y, en cualquier caso, a qué puede ser debido. Gracias:

LAZARO VELASCO
POBL. C. NUCLEAR 18 B
NAVALMORAL
CACERES

Creemos que debes ponerte en contacto con el servicio técnico de Micro Electrónica y Control, ya sea directamente o bien por medio del distribuidor donde compraste el ordenador, ya que no sabemos dónde puede residir la avería. Yo creo que es una avería que pudiera haberse originado al enchufar o desenchufar un cartucho con la alimentación del C-64 conectada, pero no puedo asegurarlo, ya que no sé si esto se ha hecho en alguna ocasión.

Que el problema varíe cuando el ordenador está "caliente" puede ser porque la CPU o los buffers de direcciones y datos, ya que la carga que soportan puede variar con la temperatura. ¡NO!... ¡No pruebes a ponerle un secador de pelo para calentarlo! la avería que podía originarse podría ser mucho peor.

EL TURBO

Me gustaría que me respondierais a unas cuantas preguntas:

—El cartucho TURBO para que el DATTASSETTE sea más veloz, ¿es útil? ¿Lo puedo adquirir en Barcelona? ¿Sirve también para cintas pregrabadas?, en definitiva, ¿vale la pena comprarlo?... Espero vuestra sincera opinión.

PEDRO CARRERAS LUQUE
C/AMPURDA, 30-2º 4ª
PARETS DEL VALLES
BARCELONA

No he tenido oportunidad de comprobar las nuevas versiones del FAST TURBO que se encuentran en el comercio, pero si son mejores que la antigua debemos descubrirnos ante los alemanes.

Hace un par de semanas me llegó una copia, de esas que circulan de uno a otro socio de los clubs de ordenadores existentes, del antiguo Fast Turbo, y he de reconocer que me dejó asombrado por la velocidad de carga de los programas, es un diez por ciento más rápido que la unidad de disco 1541 siempre y cuando el programa se encuentre justo en la vuelta donde tenemos colocada la cinta, y tengamos el turbo ya cargado.

Yo me negaba a admitirlo, pero las pruebas con cronómetro en mano son irrebatibles, claro que no todo son ventajas: Las cintas dan más problemas al tratar de leerlas con otro datassette distinto que con



el sistema original de grabación del ordenador, esto se debe a que los datos se graban sólo una vez (en vez de dos), y a más velocidad con lo que el ajuste de la cabeza debe ser perfecto.

Si de verdad merece la pena o no debes decidirlo tú, ya que con la unidad de disco puedes utilizar ficheros de acceso directo, y en cinta sólo los secuenciales... la decisión es tuya...

¿CHIP DE VIDEO ROTO?



Queridos amigos: siento no poder vanagloriarme de ser un suscriptor de su revista, pero a cambio les diré que tengo todos los números y en mi librería me los reservan todos los meses.

El motivo de escribirles, es para contarles mi problema a fin de que puedan servirse de mi experiencia todos los amigos que lo deseen.

Tengo un Commodore-64 desde hace un mes aproximadamente. Para comprarlo visité mil tiendas en busca de un precio ventajoso. Por casualidad en la tienda de un amigo mío encontré mi Commodore a un precio super fabuloso. Prácticamente a mitad de precio. Me interesé por la garantía y demás detalles por lo que pude comprobar que dicha máquina estaba totalmente en regla por lo que deduje que el precio podía ser un error de cálculo en mi amigo, pero no era así. Compre el teclado y una unidad de disco.

Cierto día, pasando un programa de su revista, se me fue la imagen de la pantalla y no pude recuperarla. Consulté en Alicante y me dijero que se había roto el chip de vídeo.

El problema vino cuando la garantía no era válida para ellos y si mandaba el aparato a su garantía de Barcelona, suponía estar en el dique seco por tres o cuatro meses. Decidí pagar la reparación en aras de la brevedad.

Lo importante de todo esto es el consejo que me dieron: NO PONER NUNCA EL ORDENADOR CERCA DE UNA FUENTE DE CALOR. Todo el mundo lo sabe, de acuerdo —NI TAN SIQUIERA BAJO LA LUZ DE UN FLEXO. Ese fue mi error. Trabajaba de noche y la luz directa de un reflector —flexo— se cargó el dichoso chip.

Ya lo sabéis amigos, NADA DE LUCES DIRECTAS. Es una pena que lo olvidasen en el manual.

Si Vd. creen que esto es verdad, sería buena idea que lo publicasen en su revista. Un saludo. ¡Ah, ánimo!.

BERNARDO BANULS SALA
AVDA. GABRIEL MIRO
EDIFICIO GLORIETA 8º-23
CALPE —ALICANTE—

Tengo mis dudas sobre la parte que pudo resultar dañada, pero me parecería más lógico si se estropease una de las ROMs al borrar su contenido por estar expuesta a algún tipo de radiación, lo que desde luego sería desaconsejable es dejar el ordenador debajo de una de las potentes lámparas de vapor de mercurio, ya que estas tienen un rico espectro de ultravioletas que podría llegar a borrar una de las memorias EPROM (que son borrables), pero creo que el 64 utiliza ROM en su lugar (estas no son borrables).

LISTAS DE PRECIOS

Apreciables amigos, mi edad es de catorce años y os escribo para pediros, si fuera posible, recibir tarifas de publicidad: Cassette, cintas, cartuchos, etc., en general de todos los accesorios del VIC-20.

La razón para que os pida ésta es que hace muy poco tiempo que tenemos el ordenador y prácticamente no sabemos nada y haber si de poco en poco podemos irnos metiendo en este pequeño gran mundo de la informática.

Ya que de momento somos unos mediocres en la dicha materia. Bueno corto el rollo y pido disculpas por las molestias que pueda ocasionar.

JOAN M. ROBLES ARANS
C/DOCTOR TRUETA, 31-33 2ª, 3ª
CASTELLDEFELS
BARCELONA

Publicamos tus señas para que las casas que distribuyen accesorios y programas para el VIC-20 te envíen las listas a tu domicilio.

SIMBOLOS RAROS

Tengo una pregunta sobre la tecla Run/Stop, y es que al colocar comillas y al apretar esta tecla, aparece en símbolo de una C invertida; ¿Cuál es el significado? ¿Para qué sirve?

Existe una tecla representada por una A minúscula dentro de un círculo, ¿para qué sirve?

Tengo un programa que al grabarlo sale en la pantalla LOAD ERROR. Existe algún POKE/S para el Commodore 64, que permita grabarlo en Cassette sin problemas, aunque en el duplicado siga apareciendo el LOAD ERROR.

Quiero advertiros que en el manual aparece en la página 21 la palabra "canviando" con "V".

ALEJANDRO CANTO NAVARRO
DOCTOR GADEA, 1 2º E y F
ALICANTE

La C invertida que aparece al pulsar la tecla de STOP cuando estás en el modo de comillas, se debe al funcionamiento interno

... PREGUNTONES ¿ ?



del ordenador que cambia la representación el trabajar entre comillas. No sé si te servirá para algo, pero de momento no le he encontrado aplicación.

La arroba que es ese símbolo que parece una "a" metida en un círculo, se usa con el programa Wedge para enviar comandos al disco. Otros programas la utilizan para enviar una ene a la impresora, otros para el PRINT AT, etc.

Respecto a la cinta, puedes leer el artículo que apareció en el número anterior sobre las "cintas rebeldes", y también un truco que hemos publicado en uno de los últimos números que te entraba el programa aunque hubiese entrado con errores.

CODIGO MAQUINA



Desearía que me explicaseis la manera de introducir programas en código Máquina mediante POKE, las direcciones de memoria que se utilizan y el significado de los números que van después de la coma.

Ruego explicación de este mi problema a algún lector compasivo (o sea buen commodoriano y amigo) a poder ser por carta, para que la explicación sea clara.

J.I.P.
C/PICO DE ARTILLEROS, 67, 1º B
28030 MADRID

Esto lo publicamos en el cursillo de lenguaje máquina del número anterior (octubre). Espero que el final del articulo conteste a tu pregunta.

HARDCOPY DE PANTALLA



Queridos editores: Me llamo Alex y ha llegado a mis manos un programa mediante el cual se puede dibujar en alta resolución con el JOYSTICK. Agradecería que publicasen o me remitiesen, si está en su poder hacerlo un programa mediante el cual pueda almacenar un dibujo en cinta magnética (cassett), mi ordenador es un VIC-20 con ampliación de 16K.

ALEX
C/ARIBAU 97, 5º 3.ª
08036 BARCELONA

En uno de los próximos números publicaremos un programa de utilidad para hacer copias de pantallas en la impresora, pero puede que no lo consigas adaptar al programa de dibujo que tienes, ya que no conozco tu programa ni por tanto la forma de hacerlo.

LOS SPRITES DEL VIC-20



Equipo de Commodore World, yo creo que para cualquier poseedor de un VIC-20 también sería positivo conocer el sistema tipo "sprite" del que no sólo goza el C-64 sino también el mencionado ordenador, aunque no lo parece, ya que en ningún libro del VIC lo pone. Por eso recurro a vosotros y en caso de ser suficientemente interesante lo aludido, deberiais publicar en algún número de vuestra revista un articulo sobre dichos sprites, por ej. cómo se definen, construyen, se hacen móviles, etc.

DANIEL GARRIGA - BARCELONA

Me has dejado asombrado con la afirmación de que el VIC tiene también el sistema de Sprites. Lamento mucho desconocer esto, y espero que cuando consigas más información, me pases lo que tengas para poder publicarlo y que todo el mundo lo sepa. Además creo que la dirección de la revista le dejaría la portada a una noticia tan sensacional...

En serio, el VIC-20 no tiene sprites, esas cosas permiten detectar colisiones en varios planos superpuestos, y lamento decirte que en el VIC no sé ningún modo de igualarlo si no es por medio de complejos programas en código máquina.

Si eres un buen programador podrás hacer que tus juegos simulen los sprites utilizando cuatro o más caracteres juntos, pero te va a resultar bastante complicado añadirle las detecciones de colisiones y todo lo demás sin perder velocidad en la ejecución del programa. Quizás te sirva de ayuda el darle un vistazo al programa "Aventuras en el Desierto", publicado en el número 4; en este artículo puedes ver cómo mover juntos varios caracteres para dibujar el jeep y el helicóptero.

LIMPIEZA DE CABEZAS

Para limpiar el datassette, qué es preferible algodón y alcohol o una cinta limpiadora, yo les pregunto: cualquier cinta limpiadora o alguna especial.

Qué cartuchos de ayuda al programador existen para el VIC-20 a excepción del Superexpander y sus precios.

LUIS JORGE LOPEZ

En principio es indiferente, pero existen algunas diferencias: la cinta limpiadora no deja residuos, y el algodón impregnado en alcohol puede dejarlos pero un poco de algodón es más barato que una cinta limpiadora, si tienes cuidado con los restos de algodón que se quedan en la cabeza, el resultado es similar y puedes utilizar los dos métodos indistintamente.

Entre los cartuchos de ayuda al programador te puedo citar sólo el que tú mencionas, el de ayuda al programador y el victree, pero los precios deberás preguntarlos a sus distribuidores respectivos ya que no los tengo (Micro Electrónica y Sakati).

El Misterio de los Pokes (VIII)



Por Diego ROMERO

Tal como hicimos en el número anterior, continuamos en este mismo número con las tablas de las direcciones de llamadas de las rutinas del intérprete Basic del Commodore 64, y en los próximos seguiremos con los demás mapas de memoria.

En este mismo número hemos publicado un artículo como complemento a esta sección, para todos aquellos que pedían que dedicásemos artículos a los mapas de memoria de pantalla del VIC-20 y del C-64, se trata de "Pokes sin dolor". El programa que aparece en ese artículo puede ser de gran ayuda a todos los que les gusta utilizar los gráficos ya incorporados al teclado y no los utilizan por no perder el tiempo en calcular los valores de los pokes que tendrían que hacer en el programa para conseguirlos. Yo mismo pienso que es bastante pesado tener que coger el manual y buscar símbolo por símbolo todos los caracteres. Espero que sirva de ayuda ese programa.

```
AD1E; Perform [NEXT]
AD78; Type match check
AD9E; Evaluate expression
AEA8; Constant - pi
AEF1; Evaluate within brackets
AEF7;    ")"
AEFF;    comma..
AF08; Syntax error
AF14; Check range
AF28; Search for variable
AFA7; Setup FN reference
AFE6; Perform [OR]
AFE9; Perform [AND]
B016; Compare
B081; Perform [DIM]
B08B; Locate variable
B113; Check alphabetic
B11D; Create variable
B194; Array pointer subrtine
B1A5; Value 32768
B1B2; Float-fixed
B1D1; Set up array
B245; "bad subscript"
B248; "illegal quantity"
B34C; Compute array size
B37D; Perform [FRE]
B391; Fix-float
B39E; Perform [POS]
B3A6; Check direct
B3B3; Perform [DEF]
B3E1; Check fn syntax
B3F4; Perform [FN]
B465; Perform [STR$]
B475; Calculate string vector
B487; Set up string
B4F4; Make room for string
B526; Garbage collection
B5BD; Check salvageability
B606; Collect string
B63D; Concatenate
B67A; Build string to memory
B6A3; Discard unwanted string
B6DB; Clean descriptor stack
B6EC; Perform [CHR$]
B700; Perform [LEFT$]
B72C; Perform [RIGHT$]
B737; Perform [MID$]
B761; Pull string parameters
B77C; Perfore [LEN]
```

```
B782; Exit string-mode
B78B; Perform [ASC]
B79B; Input byte paramter
B7AD; Perform [VAL]
B7EB; Parameters for POKE/WAIT
B7F7; Float-fixed
B80D; Perform [PEEK]
B824; Perform [POKE]
B82D; Perform [WAIT]
B849; Add 0.5
B850; Subtract-from
B853; Perform [subtract]
B86A; Perform [add]
B947; Complement FAC#1
B97E; "overflow"
B983; Multiply by zero byte
B9EA; Perform [LOG]
BA2B; Perform [multiply]
BA59; Multiply-a-bit
BA8C; Memory to FAC#2
BAB7; Adjust FAC#1/#2
BAD4; Underflow/overflow
BAE2; Multiply by 10
BAF9; +10 in floating pt
BAFE; Divide by 10
BB12; Perform [divide]
BBA2; Memory to FAC#1
BBC7; FAC#1 to memory
BBFC; FAC#2 to FAC#1
BC0C; FAC#1 to FAC#2
BC1B; Round FAC#1
BC2B; Get sign
BC39; Perform [SGN]
BC58; Perform [ABS]
BC5B; Compare FAC#1 to mem
BC9B; Float-fixed
BCCC; Perform [int]
BCF3; String to FAC
BD7E; Get ascii digit
BDC2; Print "IN.."
BDCD; Print line number
BDDD; Float to ascii
BF16; Decimal constants
BF3A; TI constants
BF71; Perform [SQR]
BF7B; Perform [power]
BFB4; Perform [negative]
BFED; Perform [EXP]
E043; Series eval 1
```

```
E059; Series eval 2
E097; Perform [RND]
E0F9; ?? breakpoints ??
E12A; Perform [SYS]
E156; Perform [SAVE]
E165, Perform [VERIFY]
E168; Perform [LOAD]
E1BE, Perform [OPEN]
E1C7, Perform [CLOSE]
E1D4, Parameters for LOAD/SAVE
E206, Check default parameters
E20E, Check for comma
E219; Parameters for open/close
E264, Perform [COS]
E26B, Perform [SIN]
E2B4, Perform [TAN]
E30E, Perform [ATN]
E37B, Warm restart
E394, Initialize
E3A2; CHRGET for zero page
E3BF, Initialize basic
E447; Vectors for $300
E453, Initialize vectors
E45F, Power-up message
E500; Get I/O address
E505, Get screen size
E50A; Put/get row/column
E518; Initialize I/O
E544, Clear screen
E566; Home cursor
E56C, Set screen pointers
E5A0, Set I/O defaults
E5B4; Input from keyboard
E632; Input from screen
E684; Quote test
E691, Setup screen print
E6B6, Advance cursor
E6ED, Retreat cursor
E701, Back into previous line
E716; Output to screen
E87C, Go to next line
E891; Perform [return]
E8A1; Check line decrement
E8B3; Check line increment
E8CB; Set color code
E8DA; Color code table
E8EA; Scroll screen
E965; Open space on screen
E9C8; Move a screen line
```

```
E9E0, Synchronize color transfer
E9F0; Set start-of-line
E9FF; Clear screen line
EA13; Print to screen
EA24; Synchronize color pointer
EA31; Interrupt - clock etc
EA87, Read keyboard
EB79; Keyboard select vectors
EB81; Keyboard 1 - unshifted
EBC2; Keyboard 2 - shifted
EC03, Keyboard 3 - "comm"
EC44; Graphics/text contrl
EC4F; Set graphics/text mode
EC78; Keyboard 4
ECB9; Video chip setup
ECE7; Shift/run equivalent
ECF0; Screen In address low
ED09, Send "talk"
ED0C; Send "listen"
ED40; Send to serial bus
EDB2, Serial timeout
EDB9; Send listen SA
EDBE; Clear ATN
EDC7; Send talk SA
EDCC; Wait for clock
EDDD, Send serial deferred
EDEF; Send "untalk"
EDFE; Send "unlisten"
EE13, Receive from serial bus
EE85, Serial clock on
EE8E; Serial clock off
EE97, Serial output "1"
EEA0, Serial output "0"
EEA9, Get serial in & clock
EEB3, Delay 1 ms
EEBB; RS-232 send
EF06; Send new RS-232 byte
EF2E; No-DSR error
EF31; No-CTS error
EF3B; Disable timer
EF4A, Compute bit count
F8D0; Check tape stop
F8E2; Set read timing
F92C; Read tape bits
FA60; Store tape chars
FB8E; Reset pointer
FB97; New character setup
FBA6; Send transition to tape
FBC8; Write data to tape
FBCD; IRQ entry point
FC57; Write tape leader
FC93; Restore normal IRQ
FCB8; Set IRQ vector
FCCA; Kill tape motor
FCD1; Check r/w pointer
FCDB; Bump r/w pointer
FD50; Initialize system constnts
FCE2; Power reset entry
FD02; Check 8-rom
FD10; 8-rom mask
FD15; Kernal reset
FD1A; Kernal move
FD30; Vectors
FD9B; IRQ vectors
FDA3; Initialize I/O
FDDD; Enable timer
FDF9; Save filename data
FE00; Save file details
FE07; Get status
FE18; Flag status
FE1C; Set status
FE21; Set timeout
FE25; Read/set top of memory
FE27; Read top of memory
FE2D; Set top of memory
FE34; Read/set bottom of memory
FE43; NMI entry
FE66; Warm start
FEB6; Reset IRQ & exit
FEBC; Interrupt exit
FEC2; RS-232 timing table
FED6; NMI RS-232 in
FF07; NMI RS-232 out
FF43; Fake IRQ
FF48; IRQ entry
FF81; Jumbo jump table
FFFA; Hardware vectors
EF59; RS232 receive
EF7E; Setup to receive
EFC5, Receive parity error
EFCA; Receive overflow
EFCD; Receive break
EFD0; Framing error
EFE1; Submit to RS232
F00D; No-DSR error
F017; Send to RS232 buffer
F04D; Input from RS232
F086, Get from RS232
F0A4; Check serial bus idle
F0BD; Messages
F12B; Print if direct
F13E; Get..
F14E,     ..from RS232
F157, Input
F199; Get.. tape/serial,rs232
F1CA; Output..
F1DD,     ..to tape
F20E; Set input device
F250, Set output device
F291; Close file
F30F; Find file
F31F; Set file values
F32F, Abort all files
F333; Restore default I/O
F34A; Do file open
F3D5; Send SA
F409; Open RS232
F49E; Load program
F5AF; "searching"
F5C1, Print filename
F5D2, "loading/verifying"
F5DD; Save program
F68F, Print "saving"
F69B, Bump clock
F6BC; Log PIA key reading
F6DD, Get time
F6E4, Set time
F6ED; Check stop key
F6FB, Output error messages
F72D; Find any tape headr
F76A; Write tape header
F7D0; Get buffer address
F7D7, Set buffer start/end
        pointers
F7EA; Find specific header
F80D; Bump tape pointer
F817, "press play.."
F82E; Check tape status
F838; "press record.."
F841; Initiate tape read
F864, Initiate tape write
F875; Common tape code
```

Pokes sin dolor

¿Te hace temblar la simple idea de tener que calcular los valores de los Pokes para los caracteres gráficos personalizados? Pues, no sufras más, este programa facilitará y acelerará tu trabajo.

Si alguna vez has diseñado una pantalla utilizando todos los gráficos del teclado de tu Commodore, seguramente habrás pasado muchas horas calculando los valores de los Pokes y las posiciones de cada carácter. El uso del papel y lápiz ayuda muy poco en la laboriosa tarea de calcular estos diseños. Existe otro método en que el ordenador, y no tú, realiza la mayor parte del trabajo.

El programa "Ayuda para el Diseño de Caracteres Gráficos" te quita el peso de tener que calcular el valor de cada carácter en pantalla, su posición en la memoria de pantalla, su valor en color y su posición en la memoria de color. Este programa resulta útil si vas a crear un diseño en pantalla mediante unos Pokes en un programa tuyo.

Existen dos versiones de este programa, uno para el VIC-2Ø (con o sin ampliación) y uno para el C-64. Dado que son prácticamente iguales, haré referencia solamente al del C-64, pero la información presentada aquí puede ser aplicada igualmente al VIC-2Ø, a no ser que se indique lo contrario.

Teclea el programa y salvarlo antes de ejecutarlo. El programa hace uso de muchos Pokes, de modo que si te equivocas al teclearlo, lo podrías perder al ejecutarlo.

Para usar este programa, cárgalo en el ordenador, pero no lo ejecutes en seguida. Una vez cargado el programa, borra la pantalla y empieza a crear tu dibujo. Puedes usar cualquier carácter del teclado (incluyendo los inversos). La única limitación es que la primera línea tiene que quedar vacía.

Cuando termines de hacer tu dibujo, mueve el cursor hasta la parte superior de la pantalla, teclea RUN y pulsa la tecla Return. Primero aparecen los titulos, y a continuación un mensaje que te preguntará si lo quieres volcar en pantalla o en impresora.

Si decides volcarlo en pantalla, necesitarás un joystick (en el PORT dos) para el desplazamiento del cursor, que debe encontrarse ahora en el centro de la pantalla. También aparecerán cinco números en la parte superior de la pantalla (cuatro en el VIC-2Ø). Estos son, en orden:

—Posición en pantalla.
—Valor del offset (esto no está presente en el VIC-2Ø).
—Código del carácter.
—Posición del color.
—Código del color.

Los números representan la posición del cursor y lo que se encontraba anteriormente en ese mismo lugar. La posición de la pantalla es la dirección de memoria donde tendrías que introducir un número mediante un Poke para que el carácter representado apareciera en la pantalla en el mismo lugar.

El valor OFFSET es el número de espacios a partir del principio de la pantalla a su posición actual.

El código de carácter es el número que representa el carácter que se encontraba antes en la posición del cursor.

La posición de color es la posición de memoria que contiene el valor del color de dicho punto en la pantalla.

Finalmente, el código de color es el color de la posición antes de la llegada del cursor.

La versión para el VIC-20 funcionará con o sin la ampliación de memoria y proporcionará las posiciones de memoria correctas tanto para la posición en pantalla como la posición de color, según la cantidad de memoria empleada.

Los números se modifican mediante el desplazamiento del cursor. Además, las posiciones quedan restauradas a su estado original en el momento en que el cursor abandona ese punto.

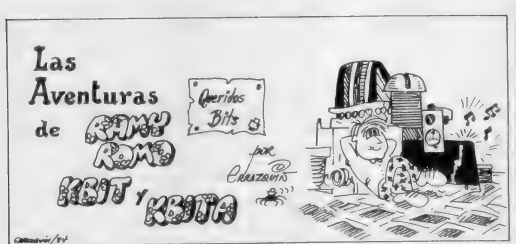
Si tienes un 1515, un 1525 u otra impresora con el interface apropiado, puedes disponer de una copia impresa de la información en pantalla.

Cuando el programa te pregunta el tipo de salida, teclea P (printer) y prepárate a descansar. El ordenador te dará información sobre cada posición siempre y cuando no contenga un espacio. La copia impresa contendrá toda la información presentada en pantalla, además del valor de color del fondo y borde de la pantalla.

Si no tienes impresora, no hace falta que teclees la parte del código dedicada a la impresora. Por lo tanto, elimina las líneas a partir de 1000, además de las líneas 220-250.

Usando este programa, hago el trabajo de una o dos horas en media hora. El programa me permite crear mis dibujos directamente en la pantalla, de modo que visualizo mejor mis ideas que cuando utilizaba solamente papel.

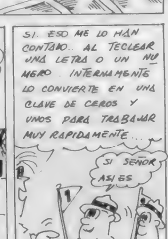
Además, con este programa no tienes que consultar libros y gráficos para obtener los mismos resultados que antes. Cuando acabas tu dibujo, los números que se necesitan para reproducir el diseño aparecen inmediatamente. Deja que tu ordenador realice las tareas pesadas mientras tú te dedicas a la creación.


Las Aventuras de RAMY ROMO KBIT Y KBITA
Queridos Bits
por Errazquin


¿?


HOLA MUCHACHOS, ME LLAMAN KBIT ¿DONDE VAIS TAN CONTENTOS Y TAN JUNTITOS?
HOLA, NOSOTROS SOMOS BITS


VAMOS DE 8 EN 8
SI, PORQUE CADA 8 DE NOSOTROS...
REPRESENTA UN CARACTER INTERNAMENTE..
UNA LETRA O UN NUMERO
ESO... NOSOTROS VALEMOS CERO O UNO
PORQUE EL EQUIPO TRABAJA BINARIAMENTE
SI TIENE ELECTRICIDAD O MAGNETISMO ES UN UNO, SI NO UN CERO
JUSTO


SI. ESO ME LO HAN CONTADO.. AL TECLEAR UNA LETRA O UN NUMERO. INTERNAMENTE LO CONVIERTE EN UNA CLAVE DE CEROS Y UNOS PARA TRABAJAR MUY RAPIDAMENTE...
SI SEÑOR ASI ES


LOS CODIGOS SON FIJOS.. ASI EL COMMODORE. Y OTROS MUCHOS TOMAN ESTA CONVERSION DE UNA TABLA INTERNA QUE SE LLAMA ASCII, DONDE ESTAN TODAS LAS LETRAS Y NUMEROS CON SU CODIGO
TODOS TODOS


VAMOS A LA MEMORIA.. Y.. REPRESENTAMOS UNA LETRA A MAYUSCULA
AUNQUE NOS PUEDEN TRANSMITIR O LEER DE UN DISCO
...EN ESTE CASO NOS HAN TECLEADO


SI... A CADA 8 SE LE LLAMA BYTE O CARACTER.. YO TENGO UN AMIGO BYTE...
ANDA QUE BIEN


DENTRO DEL ORDENADOR TODO VA EN BITS
..CLARO.. ¡ES VERDAD!
AUNQUE EXTERNAMENTE ENTRAN BYTES
..Y SALEN, POR PANTALLA E IMPRESORA
SI.. PORQUE SI NO LOS HUMANOS NO LO ENTENDERIAN, AL VER SÓLO CEROS Y UNOS


ESTO.. Y DIGO YO ¿POR QUÉ NO NOS DIVERTIMOS JUNTOS UN POCO ... SABEIS JUGAR AL CORRECALLES?
¿..AL CORREQUEEEE?


YO OS LO EXPLICARÉ... VEREIS ... BZZZ.. BISSS.. BLA, BLA, BLA ... JUMM! ...Y LUEGO.. BLA, BLA Y ...


YUPPIIIII
JI, JI, JI
OLE
VOY


BUFF... NOS HEMOS DADO UNA BUENA PALIZA
YA LO CREO.. PERO NOS HEMOS DIVERTIDO MUCHO
AHORA DEBEMOS SEGUIR NUESTRO CAMINO


OS ACOMPAÑO. HASTA LA MEMORIA
OYE .. ¿ESTA BANDERITA DE QUIEN ES?
PUES VAMONOS ENTONCES
NO SÉ YO LLEVO DOS
A LA MEMORIA


YA LLEGAMOS ¡BIEEEENNNNN!
UN MOMENTO. ALTOOOO


¿DONDE CREEIS QUE ESTAIS? TENEIS QUE ENTRAR CADA UNO CON UNA BANDERA.. Y FORMANDO UN CODIGO QUE SEA RECONOCIDO DE LA TABLA ASCII
ASI QUE LARGO DE AQUI


ES CIERTO. DEBEMOS DE IR EN FILA CON EL ORDEN QUE REPRESENTA NUESTRO CARACTER EN ASCII
¡Y NO NOS ACORDAMOS CUAL BANDERA ES Y COMO IBAMOS EN LA FILA!
BUAAAAAAA


NO OS PREOCUPEIS.. TENGO UNA IDEA.. ¡COGED UNA BANDERITA CADA UNO Y SEGUIDME!


¡!
NO ENTRAMOS
NO
VUELVO LUEGO
ADIOS
SI
NOS VAMOS


FUERAAAA.. A MI ME VAIS A ENGAÑAR CON EL VIEJO TRUCO DE ANDAR AL REVÉS.
PAF

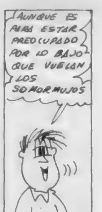

HUMM... ESTÁ BIEN LO HAREMOS DE OTRO MODO ... NECESITAMOS UN ABRIGO LARGO.. Y..
¿?


¿COMO VA LA GUARDIA MUCHACHO?
SIN NOVEDAD MI GENERAL
ERRAZQUIN/84

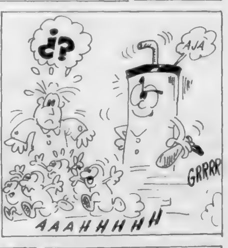

SIGA ASI MUCHACHO SIGA ASI
JI JI
¡!
.. A SUS. OR EHHHH
PAF


LARGOOOO DE AQUI TRAMPOSOOOS
ZIUUU

ESTA BIEN, LE ENTRETENDRÉ UN POCO. PARA QUE PASEIS SIN VER VISTOS.. ESTAR ATENTOS..

HOLA AMIGO... ¿COMO ESTÁS?

AUNQUE ES PARA ESTAR PREOCUPADO POR LO BAJO QUE VUELAN LOS SOMORMUJOS

POR CIERTO, POR ALLI VA UNO
¿DONDE..?
¿DONDE?
ZUOOOSSSSS

HUY. NO HE PODIDO VERLOS ¿COMO SON.. EH?
PUES JI, JI
GRRRRR
GLUP
AHHHH
ÑAC
AYYYYY
HUYYYY
SOCORROOOO

¿?
AJA
GRRR
AAAHHHHH

MUY BIEN SULTAN ..PERRITO BUENO...
AAAHHHH
GRRRR

Y DIGO YO. NO SERIA MEJOR QUE OS QUEDARAIS A VIVIR CONMIGO
NO PODEMOS
SI CUANDO APAGUEN EL EQUIPO NO ESTAMOS GRABADOS EN DISCO MORIREMOS AL NO TENER LUZ

YA ESTAAA .. AHORA SI QUE ENCONTRÉ LA SOLUCION, YO TENGO UN MANUAL DE ASCII, BUSCO LA LETRA "A", VEO EL CARACTER Y SU CÓDIGO Y...
Yupiii

VOSOTROS OS COLOCAIS EN EL MISMO ORDEN.. A VER ..TU, EL PRIMERO, LUEGO VAS TU...

AHORA SI PODEIS PASAR
ADIOOOOS AMIGUETE

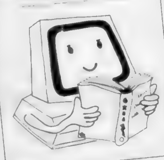
¿SABEIS UNA COSA? ES LA PRIMERA HISTORIA QUE ME HA DIVERTIDO TOTALMENTE
..O CASI
BASIC
ERRAZQUIN/84

DIRECTORIO

EXCURSION POR LA MICROPROGRAMACION

Cursillo de Lenguaje Máquina —VII—

Por Diego ROMERO



Vistos los registros de nuestras CPU's, los modos de direccionamientos que utilizan y su juego de instrucciones, es la hora de pasar a la practica, y en este número voy a explicar cómo funciona el "AUTO-RUN" que utilizaba el programa-comando OLD, tal como prometí en el numero 3.

Tambien me habeis pedido un programa desensamblador, he de aclarar que fue publicado en el numero 2 de la revista, con el nombre de "UP PERISCOPE". Este programa era además un pequeño monitor de lenguaje máquina escrito en BASIC, que permite introducir los datos hexadecimales en la memoria por medio de la opción "2-corrección" (para laterar el contenido de la memoria), cambiar de base (binaria, hexa y decimal), etc. Puede ser muy util para simplificar la introducción de programas en codigo maquina en caso de que no dispongais de otro monitor.

Como el listado aparecido en el número 3 salió bastante borroso, y para que todos lo tengáis a mano, lo reproducimos otra vez en el listado 1.

El volcado hexadecimal y desensamblado de programa generador de old, era el del listado 2. Podéis ver que llama a varias rutinas del Kernal de nuestros ordenadores (FFBA, FFBD, etc.).

Quien disponga de la guia de referencia del programador puede ver que la función de este programa es cambiar el contenido de las posiciones $0326 y $0327, que corresponden el vector de salida de mensajes, por eso debemos pulsar las teclas RECORD y PLAY antes de hacer el SYS28*256 del programa original. Después de alterar el contenido de estas dos posiciones, graba en la cinta el programa OLD, que no es otro que esos dos bytes, pero el nombre no es solamente OLD, en realidad es OLD y un montón de "cosas" que son otro programa en codigo máquina que aparece en el listado 3.

La grabación de esos dos bytes, podéis ver si miráis la guia de referencia, se realiza con dirección secundaria 1 al llamar a la rutina SETLFS con uno en el registro indice Y, esto hace que el programa OLD entre forzosamente en las posiciones originales al hacer LOAD, y este es el detalle que fuerza el auto-run, ya que cuando al final de la carga el ordenador va a sacar en pantalla el mensaje "READY", al enviar la R se encuentra con el vector que le fuerza a ir a la rutina que tenemos en el nombre del programa.

La rutina colocada en el nombre del programa vuelve a localizar los lazos de las lineas, que son los dos primeros bytes de la linea (ocultos y que no aparecen al hacer LIST) que apuntan al primer byte ocupado por la siguiente linea del programa BASIC, y asi hasta que encuentra los tres bytes que son ceros y que indican que es el final del programa, luego modifican los vectores del final del programa, que son los que ocupan las posiciones de memoria $2D y $2E, los de final de arrays, etc. y sale de nuevo al interprete BASIC.

Este es todo el secreto del comando OLD, y como veréis podeis cambiar la rutina colocada en el nombre del programa para que ejecute otra cosa, o alterar los bytes en las posiciones $0326 y $0327 para que el auto-run ejecute alguna rutina que tengáis oculta en vuestros programas, que los alteren, protejan el STOP y RESTORE, cambien de lugar una parte del programa, etc...

PROGRAMA ORIGINAL DEL GENERADOR DE OLD

```
10 poke56,28:poke55,0:clr:ad=28*256
20 readd:ifd>256then40
30 pokead,d:su=su+d:ad=ad+1:goto20
40 ifd<>suthenprint"error en datas":end
50 print"press record & play":print" y pulsa una tecla"
60 getz$:ifz$=""then60
80 sys28*256:end
100 data 169, 0, 32, 144, 255
110 data 162, 81, 160, 3, 142, 38, 3, 140
120 data 39, 3, 169, 1, 162, 1, 160, 1
130 data 32, 186, 255, 169, 104, 162, 54, 160
140 data 28, 32, 189, 255, 162, 38, 160, 3
150 data 134, 254, 132, 255, 169, 254, 162, 40
160 data 160, 3, 32, 216, 255, 32, 138, 255
170 data 96, 79, 76, 68, 32, 32, 32, 32
180 data 32, 32, 32, 32, 32, 32, 32, 32
190 data 32, 166, 43, 164, 44, 134, 254, 132
200 data 255, 160, 4, 177, 254, 240, 3, 200
```

EXCURSION POR LA MICROPROGRAMACION

Continuación Listado 1º

```
210 data 208, 249, 200, 152, 160, 0, 24, 101
220 data 254, 145, 254, 200, 169, 0, 101, 255
230 data 145, 254, 160, 0, 177, 254, 208, 30
240 data 200, 177, 254, 208, 25, 169, 2, 24
250 data 101, 254, 133, 45, 133, 47, 133, 49
260 data 169, 0, 101, 255, 133, 46, 133, 48
270 data 133, 50, 32, 138, 255, 96, 160, 0
280 data 177, 254, 170, 200, 177, 254, 133, 255
290 data 134, 254, 56, 176, 205,19158
```

DESENSAMBLADO DE LA RUTINA GENERADORA DE OLD

```
., 1c00  a9 00      lda #$00  ;esto anula los
., 1c02  20 90 ff  jsr $ff90 ;mensajes kernal
., 1c05  a2 51      ldx #$51
., 1c07  a0 03      ldy #$03  ;esto cambia el
., 1c09  8e 26 03 stx $0326 ;vector de salida
., 1c0c  8c 27 03 sty $0327
., 1c0f  a9 01      lda #$01
., 1c11  a2 01      ldx #$01
., 1c13  a0 01      ldy #$01
., 1c15  20 ba ff  jsr $ffba ;esto es SETLFS
., 1c18  a9 68      lda #$68
., 1c1a  a2 36      ldx #$36
., 1c1c  a0 1c      ldy #$1c
., 1c1e  20 bd ff  jsr $ffbd ;esto es SETNAM
., 1c21  a2 26      ldx #$26
., 1c23  a0 03      ldy #$03  ;fija la direccion
., 1c25  86 fe      stx $fe   ;de comienzo de
., 1c27  84 ff      sty $ff   ;SAVE
., 1c29  a9 fe      lda #$fe
., 1c2b  a2 28      ldx #$28  ;idem direccion
., 1c2d  a0 03      ldy #$03  ;final de SAVE
., 1c2f  20 d8 ff  jsr $ffd8 ;esto es SAVE
., 1c32  20 8a ff  jsr $ff8a ;restaura los vectores
., 1c35  60         rts       ;del sistema
.
       Lo siguiente es el nombre del programa que salvamos en cinta

.' 1c36 old    .     4f 4c 44 20 20 20 20 20
.' 1c3e              20 20 20 20 20 20 20 20
.
.
., 1c46  a6 2b      ldx $2b   ; todo esto es
., 1c48  a4 2c      ldy $2c   ; el programa
., 1c4a  86 fe      stx $fe   ; que esta oculto
., 1c4c  84 ff      sty $ff   ; en el buffer
., 1c4e  a0 04      ldy #$04  ; del cassette
., 1c50  b1 fe      lda ($fe),y
., 1c52  f0 03      beq $1c57
., 1c54  c8         iny
., 1c55  d0 f9      bne $1c50
., 1c57  c8         iny
., 1c58  98         tya
., 1c59  a0 00      ldy #$00
., 1c5b  18         clc
., 1c5c  65 fe      adc $fe
., 1c5e  91 fe      sta ($fe),y
., 1c60  c8         iny
., 1c61  a9 00      lda #$00
```

EXCURSION POR LA MICROPROGRAMACION

Continuación Listado 2º

```
., 1c63  65 ff     adc $ff
., 1c65  91 fe     sta ($fe),y
., 1c67  a0 00     ldy #$00
., 1c69  b1 fe     lda ($fe),y
., 1c6b  d0 1e     bne $1c8b
., 1c6d  c8        iny
., 1c6e  b1 fe     lda ($fe),y
., 1c70  d0 19     bne $1c8b
., 1c72  a9 02     lda #$02
., 1c74  18        clc
., 1c75  65 fe     adc $fe
., 1c77  85 2d     sta $2d
., 1c79  85 2f     sta $2f
., 1c7b  85 31     sta $31
., 1c7d  a9 00     lda #$00
., 1c7f  65 ff     adc $ff
., 1c81  85 2e     sta $2e
., 1c83  85 30     sta $30
., 1c85  85 32     sta $32
., 1c87  20 8a ff  jsr $ff8a
., 1c8a  60        rts
., 1c8b  a0 00     ldy #$00
., 1c8d  b1 fe     lda ($fe),y
., 1c8f  aa        tax
., 1c90  c8        iny
., 1c91  b1 fe     lda ($fe),y
., 1c93  85 ff     sta $ff
., 1c95  86 fe     stx $fe
., 1c97  38        sec
., 1c98  b0 cd     bcs $1c67
.
```

```
DATOS CONTENIDOS EN EL BUFFER DE CASSETTE AL CARGAR EL OLD

.
.' 0341 old        4f 4c 44 20 20 20 20 20
.' 0349            20 20 20 20 20 20 20 20
.' 0351 &+$,..>.?  a6 2b a4 2c 86 fe 84 ff
.' 0359  .1>0.hp   a0 04 b1 fe f0 03 c8 d0
.' 0361 9h. ..%>   f9 c8 98 a0 00 18 65 fe
.' 0369 .>h).%?.   91 fe c8 a9 00 65 ff 91
.' 0371 > .1>p.h   fe a0 00 b1 fe d0 1e c8
.' 0379 1>p.)..%   b1 fe d0 19 a9 02 18 65
.' 0381 >.-./.1)   fe 85 2d 85 2f 85 31 a9
.' 0389 .%?...0.   00 65 ff 85 2e 85 30 85
.' 0391 2 .?  .1   32 20 8a ff 60 a0 00 b1
.' 0399 >*h1>.?.   fe aa c8 b1 fe 85 ff 86
.' 03a1 >80m????   fe 38 b0 cd ff ff ff ff
.' 03a9            20 20 20 20 20 20 20 20
.' 03b1            20 20 20 20 20 20 20 20
.' 03b9            20 20 20 20 20 20 20 20
```

DESENAMBLADO DE LA RUTINA OLD UBICADA EN EL BUFFER DEL CASSETTE

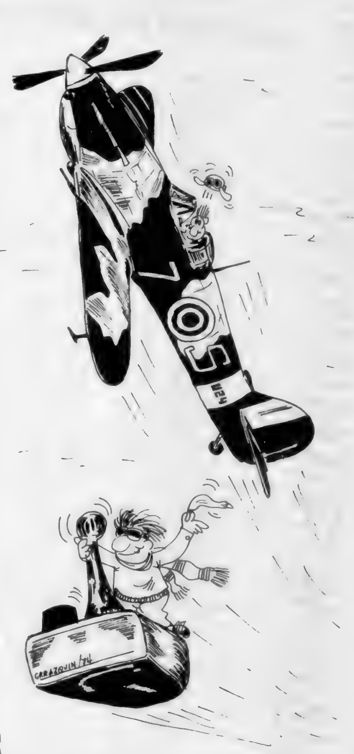
```
.
., 0351  a6 2b     lda $2b     ;lee el vector
., 0353  a4 2c     ldy $2c     ;de comienzo de
., 0355  86 fe     stx $fe     ;memoria destinada
., 0357  84 ff     sty $ff     ;a programas BASIC
., 0359  a0 04     ldy #$04    ;busca el 00 de
., 035b  b1 fe     lda ($fe),y
., 035d  f0 03     beq $0362 ;fin de linea
., 035f  c8        iny
., 0360  d0 f9     bne $035b ;sigue buscando
., 0362  c8        iny         ;hasta que lo
., 0363  98        tya         ;encuentra
., 0364  a0 00     ldy #$00    ;entonces el vector
., 0366  18        clc         ;$fe-$ff apunta
., 0367  65 fe     adc $fe     ;al lazo de la
., 0369  91 fe     sta ($fe),y  ;siguiente
., 036b  c8        iny          ;linea basic
., 036c  a9 00     lda #$00
., 036e  65 ff     adc $ff
., 0370  91 fe     sta ($fe),y
., 0372  a0 00     ldy #$00   ;comprueba si los
., 0374  b1 fe     lda ($fe),y ;dos bytes
., 0376  d0 1e     bne $0396 ;de enlace son 00
., 0378  c8        iny         ;(fin de programa).
., 0379  b1 fe     lda ($fe),y
., 037b  d0 19     bne $0396
., 037d  a9 02     lda #$02
., 037f  18        clc
., 0330  65 fe     adc $fe
., 0382  85 2d     sta $2d     ;coloca los
., 0384  85 2f     sta $2f     ;vectores de
., 0386  85 31     sta $31     ;areas de arrays,
., 0388  a9 00     lda #$00    ;strings,variables,
., 038a  65 ff     adc $ff     ;fin de programa
., 038c  85 2e     sta $2e     ;etc.
., 038e  85 30     sta $30
., 0390  85 32     sta $32
., 0392  20 8a ff  jsr $ff8a ;restaura los vectores
., 0395  60        rts         ;del sistema y fin
., 0396  a0 00     ldy #$00    ;lee el lazo
., 0398  b1 fe     lda ($fe),y ;de una linea
., 039a  aa        tax         ;a la siguiente
., 039b  c8        iny         ;para buscar
., 039c  b1 fe     lda ($fe),y ;el final del
., 039e  85 ff     sta $ff     ;programa.
., 03a0  86 fe     stx $fe
., 03a2  38        sec         ;hace que salte
., 03a3  b0 cd     bcs $0372 ;siempre sin utilizar
., 03a5  ff        ???         ;direccionamiento
., 03a6  ff        ???         ;absoluto
., 03a7  ff        ???
., 03a8  ff        ???
., 03a9  20 20 20  jsr $2020
```

VIDEO CASINO

Perdido entre las nubes

Gabe GARGIULO (RUN EEUU)

Traducido por Valerie SHANKS y adaptado por Alvaro IBAÑEZ



Tú eres el piloto de un pequeño avión de hélice, y si no se te nubla la vista, puedes aprender bastante sobre la animación en pantalla.

¿Te apetece aprender, con la ayuda de tu VIC-20 ó C-64, cómo se produce la animación en pantalla para los juegos profesionales? El juego "Perdido entre las Nubes" hace buen uso del juego de caracteres incorporado en los equipos Commodore para producir imágenes que se parecen a un monoplano, visto desde arriba. De acuerdo con la dirección en que se dirige, el avión apuntará hacia el Norte, el Sur, el Este o el Oeste.

Para jugar, necesitas un joystick para desplazar el avión por la pantalla, esquivando las nubes que no te dejan ver. Si tocas solamente una nube, el juego terminará. El juego consiste en seguir volando el máximo tiempo posible para poder superar la última puntuación.

Las nubes están apareciendo en pantalla constantemente y al azar. Cuanto más tiempo te quedes en el aire, más te costará no tropezar con las nubes. Tarde o temprano, toda la pantalla se llenará de nubes. Pero, para entonces, habrás chocado y el juego habrá terminado.

Más allá de las nubes

El avión puede salir por un lateral de la pantalla y aparecer por el otro. Sin embargo, no puede hacer esto por la parte superior ni la inferior. Si llegas a tocar esos extremos, tu avión chocará contra un muro invisible. Esta característica fue añadida para que el juego resultara más interesante. Si prefieres, puedes modificar el programa para que el avión entre y salga también por las partes superior e inferior.

VIC-20 NO AMPLIADO - C-64

(Luego te explicará cómo se hace.)

A medida que avanza el juego, se aumenta la velocidad. Al principio, un bucle de retraso, se repite de 1 a 2ØØ entre cada jugada. Es decir, que el ordenador cuenta de 1 a DLAY antes de pasar a la siguiente jugada. Sin embargo, la variable DLAY se decrementa cada vez, de modo que hacia el final el bucle hace poca cosa. El ordenador tiene que contar de 1 hasta 1, y esto, lógicamente, no le ocupa mucho tiempo.

El avión empieza a volar por el lateral izquierdo de la pantalla a un nivel elegido al azar. Se elige este nivel, F, en la línea 32Ø (la 26Ø en el listado para el Vic-20). Se llega a la posición inicial del avión, B1, multiplicando F por 4Ø (ó 22 en el caso del VIC-20) y añadiendo CHAR, el principio de la memoria de caracteres (diferente para los dos ordenadores pero correctamente definido en cada programa). Este número, F, se utiliza como el nivel elegido al azar donde el avión inicia su vuelo.

El avión empieza a volar en la línea 38Ø (la 32Ø en el listado para el VIC-20). Primero, se elige una posición al azar (I) para cada nube nueva, y una pelotita blanca, introducida mediante un Poke, aparece en la pantalla. A continuación el bucle de retraso empieza a funcionar, y se va reduciendo el tiempo de retraso.

Control del movimiento

Merece la pena hablar un poco sobre el movimiento del avión. Hemos visto cómo se desplaza un objeto por la pantalla en programas, cómo "Dibujar" por ejemplo, publicados en Video Casino. Para que un objeto se esté moviendo continuamente, siempre se añade una variable, Delta, a B1. Si no se detecta ningún movimiento del joystick, Delta siempre se queda igual, y el movimiento continúa en la misma dirección. Delta asume otro valor solamente cuando se toca el joystick.

En este juego cada nuevo valor para Delta también produce un cambio en el carácter del cursor. Cuando el avión empieza a volar, se dirige de izquierda a derecha, y el cursor tiene un valor de 1Ø7. Este carácter, una T tumbada apuntando hacia la derecha, se parece a un avión visto desde arriba, volando de izquierda a derecha.

Cuando Delta se convierte en -4Ø ó -22, lo que quiere decir que el avión sube, la variable Plane es igual a 113, y se parece a una T al revés. Cuando se dirige hacia abajo el avión tiene la forma de una T normal, mientras que se parece a una T tumbada apuntando hacia la izquierca cuando va volando de derecha a izquierda.

Esta es la base que utilizan los programadores profesionales para representar el movimiento en la pantalla, aunque también pueden existir varios caracteres para el desplazamiento en direcciones que no sean solamente Norte, Sur, Este y Oeste. Además, habrá más de uno para cada dirección para producir la sensación de un movimiento suave.

Sin embargo, para este juego nos basta con cuatro.

En la línea 56Ø (la 5ØØ en el listado para el VIC-20), el programa comprueba si el avión ha chocado con una nube. Hace un Peek para saber la siguiente posición del avión, B1, y si ésta no es 32 sabe que el avión ha chocado. Si no ha habido ningún tipo de accidente, se hace un Poke con el avión, y la posición anterior B1-Delta queda sustituida por un espacio (32).

Un choque produce los efectos necesarios visuales y de sonido, y se hace una comparación entre el momento de arranque, BEGN, y el momento del choque. La diferencia entre estas dos variables se mide en "jiffies", o intervalos de 1/60 de segundo, y el tiempo transcurrido aparece en pantalla. Si tu juego ha durado más que el tiempo indicado en la última puntuación ganadora, HS, ésta se vuelve a fijar, se anuncia un nuevo récord y te permite seguir jugando.

"Enróllate"

Te dije que te iba a dar algunas ideas sobre la forma que tiene la pantalla de "enrollarse" sobre sí misma. Ambos programas contienen líneas que no permiten que el avión pase más allá de E, el límite final de la memoria de vídeo, o de B, el principio de esta. En el listado para el C-64, por ejemplo, verás que las líneas 5ØØ y 53Ø comprueban este dato.

Si quieres modificar el programa para que el avión desaparezca por la parte superior de la pantalla y vuelva a aparecer por la parte inferior o viceversa, tienes que convertir B1 en B1-((el número de filas utilizadas en la pantalla)* (el número de caracteres en una fila -22 ó 4Ø)) siempre que B1 sea mayor que E. Si el avión se está desplazando hacia arriba, B1 se tiene que convertir en B1+((el número de filas utilizadas en la pantalla)* (22 ó 4Ø)) para que el avión aparezca por la parte inferior de la pantalla siempre que B1 sea menor que B.

Tú mismo puedes contar el número de filas utilizadas en este juego, y localizar los sitios donde tienes que hacer la modificación. No te lo voy a decir todo; algo tienes que hacer tú solito.

VIC-20 NO AMPLIADO

```
10 REM **************
20 REM *            *
30 REM *  AVIADOR   *
40 REM *            *
50 REM **************
60 PRINT"[CLR]"
70 PRINTTAB(4)"[2CRSRD][RVSON][RED][SPC]
AVIADOR[SPC][BLU][2CRSRD]"
80 PRINTTAB(2)"USA[SPC]EL[SPC]JOYSTI
CK"
90 PRINTTAB(2)"PARA[SPC]EVITAR[SPC]C
HOCAR"
100 PRINTTAB(2)"CON[SPC]LAS[SPC]NUBE
S[SPC]Y"
110 PRINTTAB(2)"GANA[SPC]PUNTOS"
120 PRINTTAB(2)"POR[SPC]CADA[SPC]SEG
UNDO[SPC]QUE"
130 PRINT"[3SPC]AGUANTES!!"
140 PRINT"[2CRSRD][3SPC][RVSON]PULSA
[SPC]UNA[SPC]TECLA"
150 GETA$:IFA$=""THEN150
160 POKE36879,234
170 PRINT"[CLR]"
180 PLANE=107
190 CSCREEN=37888+4*(PEEK(36866)AND1
28)
200 CHAR=4*(PEEK(36866)AND128)+64*(P
EEK(36869)AND120):B=CHAR:E=CHAR+484
210 DF=CSCREEN-CHAR
220 DD=37154
230 PA=37137
240 PB=37152
250 DELTA=1
260 F=INT(RND(1)*21)
270 DLAY=200
280 B1=(CHAR+F*22)-1
290 BEGN=TI
300 GOSUB560
310 GOSUB640
320 I=INT(RND(1)*484)
330 POKECHAR+I,81
340 POKECHAR+I+DF,1
350 FORJ=1TODLAY:NEXTJ
360 DLAY=DLAY-6
370 IFS0<>-1GOTO400
380 DELTA=-22:PLANE=113:IFB1+DELTA<B
THENDELTA=0
390 GOTO490
400 IFS1<>1GOTO430
410 DELTA=22:PLANE=114:IFB1+DELTA>ET
HENDELTA=0
420 GOTO490
430 IFS2<>-1GOTO460
440 DELTA=-1:PLANE=115:IFB1+DELTA<BT
HENDELTA=0
```

```
450 GOTO490
460 IFS3<>1 GOTO490
470 DELTA=1:PLANE=107:IFB1+DELTA>ETH
ENDELTA=0
480 GOTO490
490 B1=B1+DELTA
500 IFPEEK(B1)<>32GOTO650
510 POKEB1,PLANE
520 POKEB1+DF,0
530 POKEB1-DELTA,32
540 POKEB1+DF-DELTA,2
550 GOTO300
560 POKEDD,127
570 S3=-((PEEK(PB)AND128)=0)
580 POKEDD,255
590 P=PEEK(PA)
600 S1=-((PAND8)=0)
610 S2=((PAND16)=0)
620 S0=((PAND4)=0)
630 FR=-((PAND32)=0)
640 RETURN
650 FORN=1TO50
660 POKE36877,200
670 POKEB1+DF-DELTA,0
680 POKEB1+DF-DELTH,1
690 POKEB1+DF-DELTA,2
700 POKE36877,0
710 NEXTN
720 PRINT"[CLR]"
730 FSH=TI
740 ET=(FSH-BEGN)/60
750 ET=INT(ET*100)/100
760 PRINTTAB(4);"[RVSON][RED]TIEMPO:
[2CRSRD]"
770 PRINTET,"[SPC][RVSON]SEGUNDOS"
780 IFHS<ETTHENPRINT"[2CRSRD][SPC]NU
EVO[SPC]RECORD!!":HS=ET
790 FORN=1TO1000
800 NEXTN
810 PRINT"[CLR]"
820 PLANE=107
830 GOTO250
840 POKE36878,15
850 POKE36874,200
860 POKE36874,0
870 RETURN
```

C-64

```
10 REM *************
20 REM *           *
30 REM *  AVIADOR  *
40 REM *           *
50 REM *************
60 FORN=1TO10
70 READB
80 JV(N)=B
90 NEXTN
100 POKE53281,1
110 VOLUME=54296
120 VCE=54273
130 PRINT"[CLR]"
140 PRINTTAB(12)"[2CRSRD][RVSON][RED]
[SPC]AVIADOR[SPC][BLU][2CRSRD]"
150 PRINTTAB(9)"USA[SPC]TU[SPC]JOYST
ICK"
160 PRINTTAB(9)"PARA[SPC]VOLAR[SPC]E
VITANDO"
170 PRINTTAB(9)"LAS[SPC]NUBES,[SPC]Y
[SPC]CONSIGUE"
180 PRINTTAB(9)"PUNTOS[SPC]POR[SPC]C
ADA"
190 PRINTTAB(9)"SEGUNDO[SPC]QUE"
200 PRINTTAB(9)"RESISTAS!!"
210 PRINTTAB(11)"[2CRSRD][RVSON]PULS
A[SPC]UNA[SPC]TECLA"
220 GETA$:IFA$=""GOTO220
230 POKE53281,15
240 DATA-40,40,0,-1,0,0,0,1,0,0
250 PRINT"[CLR]"
260 PLANE=107
270 CSCREEN=55296
280 CHAR=1024 B=CHAR E=CHAR+484
290 B=CHAR E=CHAR+999
300 DF=CSCREEN-CHAR
310 DELTA=1
320 F=INT(RND(1)*21)
330 DLAY=200
340 B1=(CHAR+F*40)-1
350 BEGN=TI
360 GOSUB620
370 GOSUB900
380 I=INT(RND(1)*999)
390 POKECHAR+I,81
400 POKECHAR+I+DF,1
410 FORJ=1TODLAY:NEXTJ
420 DLAY=DLAY-6
430 IFDELTA<>-40GOTO460
440 PLANE=113:IFB1+DELTA<BTHENDELTA=
0
450 GOTO550
460 IFDELTA<>40GOTO490
470 PLANE=114:IFB1+DELTA>ETHENDELTA=
0
480 GOTO550
490 IFDELTA<>-1GOTO520
500 PLANE=115:IFB1+DELTA<BTHENDELTA=
0
510 GOTO550
520 IFDELTA<>1GOTO550
530 PLANE=107 IFB1+DELTA>ETHENDELTA=
0
540 GOTO550
550 B1=B1+DELTA
560 IFPEEK(B1)<>32GOTO690
570 POKEB1,PLANE
580 POKEB1+DF,0
590 POKEB1-DELTA,32
600 POKEB1+DF-DELTA,2
610 GOTO360
620 JV=PEEK(56320)
630 FR=JVAND16
640 JV=15-(JVAND15)
650 TA=JV(JV)
660 IFTA=0THENRETURN
670 DELTA=TA
680 RETURN
690 FORN=1TO50
700 POKEVCE,15
710 POKEVOLUME,15
720 POKEB1+DF-DELTA,0
730 POKEB1+DF-DELTA,1
740 POKEB1+DF-DELTA,2
750 POKEVCE,0
760 POKEVOLUME,0
770 NEXTN
780 PRINT"[CLR]"
790 FSH=TI
800 ET=(FSH-BEGN)/60
810 ET=INT(ET*100)/100
820 PRINTTAB(10);"[RVSON][RED]TIEMPO
:[2CRSRD]"
830 PRINTTAB(8)ET;"[SPC][RVSON]SEGUN
DOS"
840 IFHS<ETTHENPRINTTAB(8)"[2CRSRD]N
UEVO[SPC]RECORD!!":HS=ET
850 FORN=1TO1000
860 NEXTN
870 PRINT"[CLR]"
880 PLANE=107
890 GOTO310
900 POKEVOLUME,15
910 POKEVCE,32
920 POKEVOLUME,0
930 RETURN
```

MARKETCLUB

Servicio gratuito para nuestros lectores particulares.
De no especificar duración, los anuncios serán publicados durante 1 mes.
Debido a ciertos abusos que nos han sido comunicados, los anuncios gratuitos de Market club solamente serán publicados si vienen con nombre, apellidos y dirección completa. No se admitirán apartados postales.
Lamentamos que todos tengan que sufrir esta medida por culpa de algunos "*listos*"
Adicionalmente es recomendable que enviéis el número de teléfono por si hubiese alguna pega, si no deseáis que lo publiquemos podéis decirlo en la carta.

MERCADILLO

• Vendo VIC-20, con Datassette, guía de referencia y varios juegos. El VIC es de noviembre del 83 y lo demás de enero del 84. Todo por 41.000 ptas. Llamar al 3570429 de Barcelona. Germán Costa ***(Ref. M-135).***

• Estoy interesado en el intercambio de programas para el C-64 (juegos, utilidades, música, dibujo...). Tengo programas muy buenos. Si te interesa, llámame o escríbeme. Octavio Pérez. C/Canfranc, 9-Esc. Izq. Tel.: (976) 216891. 50004 Zaragoza. ***(Ref. M-136).***

• Vendo Commodore 64, comprado en noviembre del 83. Incluyo: datassette, manual, guía de referencias (libro de 3.000 ptas.), Joystick, suscripción de un año a las dos revistas especializadas en CBM y multitud de programas de aplicaciones y juegos. Teléfono: 4338825. J. F. Fernández Arévalo. Avda. Mediterráneo, 39-3º 1ª. 28007 Madrid. ***(Ref. M-137).***

• Desearía intercambiar programas para el Commodore-64, particularmente de aplicaciones, utilidades y juegos. Interesados, escribir a: José Luis Pérez Cortés. C/Navas de Tolosa, 32, 1º A. 50010 Zaragoza. Mandando lista de programas. ***(Ref. M-138).***

• Intercambio programas de cualquier tipo para CBM-64 tanto en disco como en cassette. Desearía contactar también con algún poseedor de compilador Pascal. O con personas interesadas en conseguirlo. Luis París García. Ctra. Vic. 119-121. 5º 2.ª Manresa - Barcelona. Teléf. 8730829 ***(Ref. M-139).***

• Deseo comprar la unidad disco VC-1541, así como los diskettes que me pudieran proporcionar quien vendiese este periférico, pero eso sí, en perfectas condiciones de funcionamiento. Las ofertas me las pueden enviar a la siguiente dirección. José M.ª Lozano Gallardo. Avda. de los Deportes, 42, 9º B. Móstoles. Madrid. ***(Ref. M-140).***

• Vendo VIC-20 + introducción al lenguaje de programación Basic I y II con cintas + cartuchos super Expander con 3K + Cartucho ajedrez "Sargón II Chess" por 45.000 pesetas. Llamar tarde. Teléf.: (967) 298282. Preguntar por Ernesto. ***(Ref. M-141).***

• Vendo VIC-20 comprado en mayo de 1983 con las 4 cintas + ampliación por 25.000 pesetas. Teléf.: (93) 8513636. Preguntar por Ramón. ***(Ref. M-142).***

• Vendo VIC-20 nuevo a estrenar a buen precio, llamar al teléfono: 8995153. Josep M.ª Carbonell Casanovas. C/Nov. 48. S, Pere de Riudevilles. Barcelona. ***(Ref. M-143).***

• Vendo programador universal de memorias EPROM para VIC-20 y CBM, graba las memorias siguientes, 2516-2716-2532-2564-2764-27128 y futuras. Félix Portabella Padro. C/Forn de Sta. Lucía, 1, 2º. Manresa. ***(Ref. M-144).***

• Vendo VIC-20 prácticamente nuevo (comprado en enero de 1984), 1 curso de Basic, cartuchos de juegos, juegos en cassette, libros y cassette por 39.000 ptas. Carlos Guardia. C/Córcega, 110, 1º 2.ª 08029. Teléf. 2397710. ***(Ref. M-145).***

• Vendo dos libros para el VIC-20: GAMES FOR YOUR VIC-20. 123 páginas (850). EXPLORING ADVENTURE GAMES ON THE VIC-20. 243 páginas (1.850). Andrés Portas Afenjar. Ríos Rosas, 5, 4ºD 29007 Málaga. ***(Ref. M-146).***

• Vendo VIC-20 en perfecto estado + Datasette C2N + Manual del Usuario + 1.ª parte del curso introducción al Basic + Cartucho (Omega Race) + cinta de indescomp (scramble) + Joystick anatómico de espectravideo + 12 revistas + cintas con juegos y programas de utilidad y vírgenes, todo ello por 31.900 ptas. Fco. Javier. Batanero del Amo. C/República Argentina, n. 62, 2º 4.ª Cornellá (Barcelona). Teléf.: 3763140 Prefijo 93. ***(Ref. M-147).***

• Intercambio juegos en cassette para C-64. Teléf. (93) 8032179. Bernat Planas Vives, Passeig Mn. Cinto Verdaguer 100 Atc. Igualada (Barna). ***(Ref. M-148).***

• Intercambio programas de juegos y aplicaciones para ORIC-1 Roger Noguera. C/Llessamins 10, Igualada (Barcelona). ***(Ref. M-149).***

• ¡Atención distribuidores! Deseo recibir información o catálogo sobre programas de juegos en cassette. A ser posible de venta en la provincia de Barcelona. Bernat Planas. Passeig Verdaguer, 100. Igualada (Barcelona). ***(Ref. M-150).***

• Vendo impresora nueva para C-64, 10 tipos de letra, gráficos, caracteres programables, etc. Luis Sánchez Pérez. Apartado nº 176. 45080 Toledo. Teléf.: 221494. ***(Ref. M-151).***

• Vendo Commodore-64 en perfecto estado (menos de un año de uso) junto con un gran número de programas: Interesados escribir a José Francisco Aldana Montes, Beethoven 6, 6º D 29004 Málaga o llamar al 317696 después de las 10. ***(Ref. M-152).***

• Urge vender Commodore 64, cassette, guía del programador, juegos, regalaría revistas, libros. Precio a convenir, también vendo por separado. Antonio Cebrián Martínez. C/Francisco Pizarro, 73. 02004 Albacete. Teléf.: (967) 234108 de 2 a 3,30 ***(Ref. M-153).***

• Vendo VIC-20, Cassette C2N, Cartucho de memoria de 16K, Super Expander, todo esto por 39.000 ptas. Los interesados llamar al (93) 699 58 80 y preguntar por José Navas. Juan José Navas Fuentes. C/Juan Ramón Jiménez, nº 26, 6º 3.ª de Rubí (Barcelona). ***(Ref. M-154).***

• Vendo monitor Commodore de Color, nuevo con garantía 60.000 ptas. Javier Taudes C/Corazón de María, 55. Madrid. Teléf. (91) 4167385 ***(Ref. M-155)***

• Intercambio programas para el Commodore 64 en cinta y disco, utilitarios y juegos. José M. Suárez., Rue J. Jeanfils nº 6. Soumagne. Belgique. ***(Ref. M-156).***

• Vendo Commodore Pet 2001 con interface para RTTY-CW, y programas por 50.000 ptas. Teléfono (942) 223985 de Santander. José Manuel ***(Ref. M-157).***

• Atención VICiosos: 1º Vendo Módulo de expansión de 16 K (para VIC-20). Intercambiaría juegos en cassette. Llamar o escribir a: Daniel Garriga. C.N. II, Waldem-7, 2-51. Sant Just Desvern (Barcelona). Tel.: (93) 3720243 ***(Ref. M-158).***

• Vendo Guía del Usuario para VIC-20 (1.500 ptas.) Guía de Referencia del Programador (1.500 ptas.) las dos partes del Curso Basic (1.700 cada uno; incluye cintas) y un cartucho de Ajedrez por 2.000 ptas. Todo por sólo 7.000 ptas. Salvador Tudela Edo. Gr. Carlos III, 130 "A". Barcelona 08034 - Teléf.: (93) 2041823 ***(Ref. M-159).***

• Soy usuario de un C-64. Desearía intercambio de información y programas de todo tipo (en cassette). Interesados dirigirse a: Marcel Planagumá. Pge. Mir i Borrell, 1-1º 2.ª Mataró (Barcelona). Tel. (93) 7985127 ***(Ref. M-160).***

CLUBS

• Club de amigos. Foto Estudio 2. Plaza de Sombrereros, 2. **Palma de Mallorca.** Tel.: 21 31 62. ***(Ref. C-1)***

• Club de programación Alaiz. Tels.: 254 480-257 704. Pamplona. ***(Ref. C-4)***

• Club de Commodore de Albacete. Fernando Martínez Guerrero. La Roda, 39. 5º D. Albacete. ***(Ref. C-5)***

• Círculo de Durensan Vigués. Vigo. C/Venezuela, 48. Entrechan. Tels.: (986) 410 683/422 519. ***(Ref. C-6)***

• Asociación Manchega de Usuarios de CBM-64 (A.M.U. C-64). C/Blasco de Garay, 10. Albacete. ***(Ref. C-13).***

• Club MICRO-ESPLAI. Gran Vía, 682. Pral. (Entre Bruch y Gerona) de Barcelona. ***(Ref. C-14).***

• Interesados en la formación de un club de informática en Granada capital diríjanse por correo a C/ San Jerónimo, 23, 2º. 18001 Granada. Miguel A. Puerta Carrasco. ***(Ref. C-28).***

• Si queréis que hagamos un club "Vic-20" poneros en contacto con Antonio Recober. Teléf.: 32 20 46. Málaga. Se trata de intercambiar programas. Conocimientos y experiencia sin que nos cueste dinero. Si algún 64 quiere hacer el camino a nuestro lado. Podemos intentarlo. ***(Ref. C-29).***

• Desearía contactar con interesados en formar un club de usuarios del C-64 en Granada e intercambiar programas y experiencias. Antonio Toral Peinado. C/Puente nº 3. Illora (Granada). Tel.: 958-46 30 09 ***(Ref. C-30).***

• Para formación club nacional de CBM Series 2000 y 3000 exclusivamente, ponerse en contacto por carta con Artemio González Pérez. Fernando Arocena Quintero, 1. Teléf.: 22 42 44. 38009 Santa Cruz de Tenerife. ***(Ref. C-31).***

• Me gustaría contactar con alguien de Motril para compartir experiencias y/o formar un club de C-64 (y VIC-20 si hace falta). Ya tengo local. Mi teléfono: 958-600 314 ***(Ref. C-32).***

• Ha nacido el club "Brian & Chip" para usuarios del C-64 y Spectrum 16K. Crearemos una revista propia, pero necesitamos colaboraciones, programas e ideas. Intercambiamos programas listados o en cinta, escribir desde cualquier rincón de España, prometo contestar. Francisco Reig López, C/Marina, 261, 3º 5ª, 08025 Barcelona. ***(Ref. C-33).***

• Agradecería la dirección o teléfono de algún socio de mi ciudad o alrededores, a fin de ponerme en contacto con él para ver la posibilidad de formar un club o similar. José Esteban Llusar Molés. Daoiz, 2-1º 2ª. Pto de Sagunto (Valencia). ***(Ref. C-34)***

• Grupo de usuarios de Commodore en Tarragona, estamos intentando formar un club lo más sólido posible, no nos mueve ningún interés lucrativo, y estamos desarrollando un sinfín de actividades, encaminadas a tener un más amplio conocimiento de nuestro ordenador. Alfred Blasi. Tel.: (977) 547 307. Tarragona. ***(Ref. C-35).***

• Deseo contactar con niños-as que deseen formar un club de ordenadores Commodore 64 y VIC-20 (puede ser otra marca), para intercambiar experiencias. C/Avda. del Norte nº 8 atc. 3ª Hospitalet (Barcelona). Manuel González Martínez. Tel.: 93-240 23 50. ***(Ref. C-36).***

• Agradecería mucho a todo el que pudiera informarme sobre la creación de un club para VIC-20 en Málaga. ***(Ref. C-37)***

• Estamos montando una actividad de informática en el Club Los Chopos con la idea de enseñar a chavales de BUP y últimos cursos de EGB a programar y formar un grupo de amigos para intercambiar programas y experiencias, sin ningún fin lucrativo. Javier Huguet. C/Antonio Menchaca, 19. Teléf.: 4693461. Neguri. ***(Ref. C-38).***

• Me gustaría contactar con interesados en la confección de programas de astronomía. Ordenador VIC-20. Interesados, escribir a: Apartado 2309. 15080 La Coruña. ***(Ref. C-39).***

N.R.: Para continuar saliendo en el futuro, envíanos Dirección.

• Los interesados en programas (VIC-20) relacionados con la frecuencia de 27 Mhz escriban a: Apartado Postal 471. 14080 Córdoba. ***(Ref. C-40).***

• Estamos formando un Club de Usuarios en el centro de Madrid. Cualquier interesado que se ponga en contacto con nosotros en el teléfono 2766694, Lunes-Miércoles-Viernes de 18,00 a 20,00 h. Dirigiéndose a Rafael (EB4XD). (Dejar teléfono en el contestador). ***(Ref. C-41).***

N.R.: Para continuar saliendo en el futuro, envíanos Dirección.

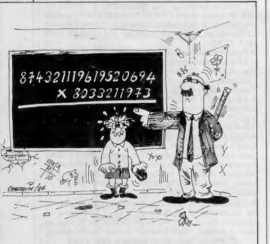
• Me gustaría contactar con usuarios de la zona de Mataró. Dirigirse a: Marcel Planagumá. Pge. Mir i Borrell, 1-1º 2ª. Mataró (Barcelona). Tel. (93) 7985127. ***(Ref. C-42).***

BOLSA DE TRABAJO

• Profesor de inglés, trabajando en un instituto y licenciado en Historia, daría clases particulares de inglés. Dirigirse a: Michael Bruggeman. Teléf.: (91) 2271143. ***(Ref. BD-1).***

Multiplicación larga en Basic

(Viene de pág. 29)



y A para el producto. El listado permite que la entrada de dos números tenga una longitud de hasta 7Ø dígitos, con el resultado de que la respuesta nunca tendrá una longitud de más de 14Ø espacios.

Al considerar la entrada, hay que tener en cuenta varios casos especiales: uno o ambos números pueden ser negativos, uno o ambos números pueden ser cero y un número, por descuido, puede contener un carácter no numérico.

En seguida que se introduce un número, el programa comprueba para ver si dicho número es negativo. Si lo es, se le quita el signo de menos y la variable FL se incrementa por 1. Al acercarse al final del proceso, en la línea 54Ø, si FL es 1 (es decir, si solamente uno de los dos números de entrada es negativo), un signo de menos se añade a la respuesta.

A continuación, se comprueba para ver si cada espacio es un dígito y no otro tipo de carácter. Si se encuentra otro carácter, el programa da un mensaje de BAD INPUT (ENTRADA INCORRECTA), y termina. De no ser así, el programa utiliza la función MID$ para dividir la cadena en carácter de dígitos individuales, realiza la función VAL para convertir los caracteres en números que pueden ser manipulados de forma matemática y los almacena en los "arrays" M1 y M2.

Hay que realizar una prueba adicional antes de introducir el bucle principal del programa, en la línea 217, comprueba para ver si una o las dos entradas es un cero. De ser así, se salta por encima del bucle principal y la respuesta, cero, se representa directamente.

Mediante el uso de los bucles For... Next en las líneas 31Ø-37Ø, los "arrays" se procesan como si los números que representan se multiplicasen en papel, con el multiplicando (el número superior) encima del multiplicador (el número inferior). El número superior se multiplica de forma secuencial, desde las unidades situadas a la izquierda, por las unidades, decenas, cientos, etc., situadas como el número inferior. Si el resultado de cualquier multiplicación es 1Ø ó más, el lugar de las decenas se lleva al siguiente lugar hacia arriba de la respuesta.

Si estuvieras realizando una multiplicación a mano, una vez multiplicado el número superior por el que se encuentra en la extrema izquierda del número inferior, tendríais que sumar los resultados de cada secuencia de multiplicaciones; pero el programa va sumando a medida que avanza. Ahora hay que sacar el resultado.

Se presentan dos problemas para sacar el resultado. Primero, sería interesante poder evitar los ceros de entrada (es decir, dos por dos debe dar 4 y no 139 ceros seguidos de un 4). Segundo, formaremos una cadena (concatenar) con nuestra respuesta a partir de los elementos individuales del "array" A, donde cada elemento es un número real de Ø a 9. La conversión de un número real de x-lugar en una cadena resulta en una cadena con una longitud de lugares de x+1; el carácter de entrada es un carácter en blanco (que significa un signo de más) o un signo de menos. Las líneas 51Ø y 52Ø se encargan de estos problemas.

El programa realiza la función que le fue asignada. Los números moderadamente cortos (de 5-7 dígitos) se multiplican relativamente rápido. Sin embargo, cuanto más largos los factores, más tarda el programa en ejecutarse. Cuando ejecute el programa para multiplicar dos números de 7Ø dígitos (el caso más extremo), el ordenador tardó más de veinte minutos en sacar el resultado. Una muestra se presenta en la Figura 1.

The VIC-20 takes 2Ø minutes to multiply two 7Ø-digit numbers with this program. About how long would it take you to do it by hand?

= Los equipos Commodore tardan viente minutos en multiplicar dos números de 7Ø dígitos con este programa. ¿Cuánto tardarías tú en hacerlo a mano?

PRIMER NUMERO? 123456789
SEGUNDO NUMERO? 111111111
EL PRODUCTO ES 1371742Ø98628257
PRIMER NUMERO? 123456789123456789123456789123
SEGUNDO NUMERO? 9876543210Ø9876543210Ø9876543210Ø
EL PRODUCTO ES 12193263124676116335924401656991921950875735405064116750483Ø

COMMENTARIOS COMMODORE WORLD

Vietronic: El 64 director de orquesta

Otro impacto lo produjo Vietronic, S. A. de Barcelona, que junto con Siel-Italia ha desarrollado un programa y unos interfaces que permiten conectar al C-64 toda una gama de instrumentos musicales electrónicos interminables: sintetizadores, órganos, etc. Todo un avance para nuestros 64 que los melómanos agradecerán.

En realidad este último avance se ha hecho posible gracias a que las grandes casas (Hammond, Roland, Korg, Yamaha, etc.) han decidido adoptar un estándard para las conexiones entre sus secuenciadores, e instrumentos, permitiendo una gran flexibilidad en la elección de instrumentos, y además se ha realizado el sueño de conectar nuestro C-64 controlándolo todo. Puedo garantizar que el resultado era super-fabuloso dentro de la sala de demostración, donde pudimos escuchar al maestro de música llamado C-64, ejecutando perfectamente toda una serie de piezas que podrían haber dejado pasmados a más de uno de nuestros grandes compositores (Imaginad las obras que podrían haber compuesto si hubieran despuesto de un C-64 y todos estos instrumentos en su época). También han



lanzado otro interface para el Spectrum de 48K, pero con bastante más limitaciones...

Las señas de Vietronic son:
Bolivia, 239
Barcelona-20

PARA LLEGAR MAS LEJOS...

SPECTRAVIDEO™



MK II







MICROSOFT BASIC ★ FORTRAN 80 ★ Microsoft Logo ★ **COLECO cartuchos**
CP/M ★ Cobol 80 ★ MACROENSAMBLADOR Z80 ★ **Ensamblador CP/M**

• **MSX. Es el sistema escogido por 16 fabricantes japoneses de ordenadores** (SONY, PANASONIC, PIONER, SANYO, SPECTRAVIDEO, etc.) **siendo compatibles todos los programas creados por cada uno de los fabricantes.**

CARACTERISTICAS GENERALES	SPECTRAVIDEO 328	SPECTRAVIDEO 318	APPLE II e	COMMODORE 64	ATARI 600 xL	BBC MODELO B	DRAGON 32	ORIC ATMOS
PROCESADOR	Z80 A	Z80 A	6502	6510	6502	6502	6809 E	6502
RELOJ EN MHz	4 MHz	4 MHz	1 MHz	1 MHz	1.8 MHz	2 MHz	1 MHz	1 MHz
MEMORIA STANDARD EN RAM	80 K	32 K	64 K	64 K	16 K	32 K	32 K	48 K
AMPLIABLE HASTA	144 K	144 K	—	—	64 K	—	64 K	64 K
MEMORIA STANDARD EN ROM	32 K	32 K	16 K	20 K	24 K	16 K	16 K	16 K
AMPLIABLE HASTA	96 K	96 K	—	—	—	—	—	—
MICROSOFT BASIC EXTENDIDO	SI	SI	—	—	—	NO	SI	SI
NUMERO DE TECLAS	87	51	62	66	57	73	53	57
TECLADO CON GRAFICOS PREDEFINIDOS	SI	SI	—	SI	SI	SI	NO	NO
TECLAS DE FUNCION PROGRAMABLES	10	10	—	8	4	10	NO	NO
MANDO JOYSTICK INCORPORADO	NO	SI	—	NO	NO	NO	NO	NO
COLORES	16	16	16	16	16	16	9	8
SPRITES	32	32	—	8	8	—	?	—
RESOLUCION (Puntos de pantalla)	256×192	256×192	280×192	320×200	320×192	256×640	256×192	240×200
TEXTO EN PANTALLA	40×24	40×24	40×24	40×25	40×24	40×32	32×16	40×28
CANALES DE SONIDO	3	3	1	3	4	1	3	3
OCTAVAS POR CANAL	8	8	4	9	?	3	5	8
A.D.S.R. ENVOLVENTE	SI	SI	NO	SI	NO	SI	NO	SI
CAPACIDAD UNIDAD DE DISCO	256 K	256 K	140 K	170 K	127 K	100 K	?	?
CENTRONICS	SI (opcional)	SI (opcional)	SI (opcional)	—	SI (opcional)	SI (opcional)	SI	SI
RS232	SI (opcional)	SI (opcional)	SI (opcional)	—	SI (opcional)	SI (opcional)	—	—
80 COLUMNAS	SI (opcional)	SI (opcional)	SI (opcional)	SI (opcional)	SI (opcional)	—	—	—
PRECIO P.V.P.	**76.000**	**49.900**	**166.618**	**79.900**	**58.500**	**140.000**	**67.800**	**59.500**

(TM)

— MSX y MICROSOFT son marcas registradas por MICROSOFT CORPORATION.
— CPM es una marca registrada por DIGITAL RESEARCH INC.
— COLECO VISION es una marca registrada por COLECO INDUSTRIES INC.



HARDWARE
DISTRIBUIDOR OFICIAL DE SPECTRAVIDEO INTERNATIONAL LTD
PASEO DE LA CASTELLANA, 179 - MADRID-16
TELEFONO (91) 279 31 05

DELEGACION EN CATALUÑA: ACE, S.A.
Tarragona, 100 - Barcelona-15.
Teléfono (93) 325 10 58

YA DISPONIBLE EN:



... Y EN TODAS LAS TIENDAS ESPECIALIZADAS.